

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра автоматизованих систем обробки інформації і управління

УДК: 004.657

«До захисту допущено»

В.о. завідувача кафедри

(підпис) О.А.Павлов
(ініціали, прізвище)

“ ” _____ 2019 р.

Дипломний проект
на здобуття ступеня бакалавра

з напрямку підготовки 6.050101 «Комп'ютерні науки»

на тему: *«Комплекс задач аналізу ринку праці фахівців з інформаційних технологій»*

Виконав:

студент 4 курсу, групи ІС-51

Пономаренко Артем Олександрович
(прізвище, ім'я, по батькові)

(підпис)

Керівник

доц., к.т.н., доц. Звенігородський О.С.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

**Консультант з
графічної
документації**

ст. викладач Москаленко Н.В.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент

проф. каф. мікроелектроніки,
к.т.н., доц. Орлов А.Т.
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проекті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент Пономаренко А.О.

(підпис)

Київ – 2019 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з п'яти розділів, містить 19 рисунків, 14 таблиць, 1 додаток, 18 джерел.

Дипломний проект присвячений розробці комплексу задач аналізу ринку праці ІТ індустрії.

Інформаційного забезпечення - описані вхідні та вихідні дані програми, структура БД.

Математичного забезпечення - обґрунтування вибору методів вирішення задачі та їх пояснення.

Програмне забезпечення - опис структури програми яку було створено для вирішення задачі.

технологічний розділ - описання керівництва користувача та проведення випробувань програми.

ЧАСОВИЙ РЯД, НЕЙРОННІ МЕРЕЖІ, СИСТЕМА АНАЛІЗУ,
ПРОГНОЗУВАННЯ

					ДП ІС-5119.1181-с.ПЗ				
		Прізвище	Підпис	Дата					
Розроб.		Пономаренко А.О.			Комплекс задач аналізу ринку праці фахівців з інформаційних технологій	Літ.	Лист	Листів	
Перевірив.		Звенигородський О.С.					2	72	
						КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51			
Н. кон.		Москаленко Н.В.							
Затв.		Павлов О.А.							

ABSTRACT

Structure and scope of work. Explanatory note of the diploma project consists of five sections, contains 28 drawings, 15 tables, 1 application, 18 sources.

The diploma project is devoted to the development of a complex of tasks for labor market analytics in IT industry.

The information support section describes the input and output data of the program, the structure of the database.

The mathematical resources section is devoted to the justification of the choice of methods for solving the problem and their description and explanation.

The software section describes the structure of the program that was created to solve the problem.

The technology section describes the user's manual and program tests.

TIME SERIES, NEURAL NETWORKS, SYSTEM OF ANALISIS, FORECAST

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ**

БД	База даних
ЗП	Заробітна плата
ОС	Операційна система
MVC	(Model-View-Controller) Модель-Представлення-Контроллер
JPA	(Java Persistence API) Специфікація API Java EE
SQL	(structured query language) Мова структурованих запитів
IDE	(Integrated development environment) Інтегроване середовище розробки
ІТ	Інформаційні технології
ДСТУ	Державні стандарти України
ГОСТ	(Государственный стандарт) Державний стандарт

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	3
ВСТУП	6
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	8
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА.....	8
1.1.1 Опис процесу діяльності	10
1.1.2 Опис функціональної моделі.....	10
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ	11
1.3 ПОСТАНОВКА ЗАДАЧІ.....	14
1.3.1 Призначення розробки	14
1.3.2 Цілі та задачі розробки.....	14
Висновок до розділу	15
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	16
2.1 ВХІДНІ ДАНІ	16
2.2 ВИХІДНІ ДАНІ	17
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ	18
Висновок до розділу	19
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....	20
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ.....	20
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ.....	20
3.3 ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ’ЯЗАННЯ	21
3.4 ОПИС МЕТОДУ РОЗВ’ЯЗАННЯ	23
Висновок до розділу	29
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	30
4.1 ЗАСОБИ РОЗРОБКИ	30

4.2 Вимоги до технічного забезпечення35

4.3 Архітектура програмного забезпечення36

 4.3.1 Діаграма класів36

 4.3.2 Діаграма послідовності38

 4.3.3 Діаграма компонентів.....38

 4.3.4 Специфікація функцій.....39

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ44

5.1 Керівництво користувача44

5.2 Випробування програмного продукту46

 5.2.1 Мета випробування46

 5.2.2 Загальні положення46

 5.2.3 Результати випробувань47

ЗАГАЛЬНІ ВИСНОВКИ49

ПЕРЕЛІК ПОСИЛАНЬ50

ВСТУП

На сьогоднішній день в Україні професія в галузі інформаційних технологій – одна з найпривабливіших, та з року в рік її вклад в економіку країни продовжує зростати. Крім того, в галузь ІТ входить велика кількість різноманітних професій, і з кожною можуть бути пов'язані специфічні вимоги до навичок працівника. Такими вимогами можуть бути володіння однією з численних мов програмування, різні рівні знання англійської мови, рівні освіти та досвід роботи. Оволодіти всіма можливими навичками що дозволять працевлаштуватися у будь-якій професії ІТ просто неможливо, тому програма яка може допомогти людям зробити вибір професії та конкретних навичок для освоєння, аргументуючи його актуальною статистичною інформацією буде дуже корисною. Саме тому програма що дозволить спростити та візуалізувати складну для розуміння статистичну інформацію яка стосується ринку ІТ, має актуальність на сьогоднішній день. Мета дослідження – спростити користувачу пошук ІТ професії шляхом розробки комплексу задач аналізу ринку праці фахівців ІТ і інструменту візуалізації показників середньої ЗП з результатів опитувань працівників ІТ, доступних в мережі Інтернет.

Основні завдання дослідження:

- розробити структуру даних для зберігання результатів опитувань;
- обґрунтувати вибір засобу прогнозування;
- описати принцип роботи обраного засобу прогнозування;
- описати програмні інструменти що використовувались в створенні програмного продукту;
- навести керівництво користувача та провести випробування програмного продукту.

Об'єкт дослідження – стан економічного положення ІТ індустрії за останнє десятиріччя.

Предмет дослідження – середні ЗП фахівців з ІТ за останнє десятиріччя.

Короткий зміст розділів пояснювальної записки дипломної роботи:

- у розділі ЗАГАЛЬНІ ПОЛОЖЕННЯ описується поточний стан ІТ індустрії, ставиться задача та розглядаються наявні засоби її вирішення;
- у розділі ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ описана структура даних з якими працює програма;
- у розділі МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ обирається математичний метод розв'язання задачі та детально описується;
- у розділі ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ описуються програмні інструменти що використовувались для створення програмного продукту та його архітектура;
- в ТЕХНОЛОГІЧНОМУ РОЗДІЛІ наведено керівництво користувача та проведення випробувань програмного продукту.

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

На сьогоднішній день в Україні професія в галузі інформаційних технологій – одна з найпривабливіших, та з року в рік її вклад в економіку країни продовжує зростати. В 2018-му році українські програмісти зайняли другу позицію серед експертних послуг, обігнавши навіть одну з найпотужніших сфер української економіки – трубопровідний транспорт.

Понад 20% усіх експортованих послуг, які надає Україна, належить ІТ індустрії. Тому іноземні компанії активно заходять на Український ринок, а вітчизняні спеціалісти складають здорову конкуренцію на світовому ринку, крім того економічна ситуація країни дозволяє більш розвиненим країнам, таким як США, Німеччині, Великій Британії та іншим наймати відносно дешеву, але спроможну та гарно освічену робочу силу.

За даними Державної фіскальної служби, сума податків сплачених ІТ-компаніями податків у 2014-2018 роках щорічно збільшувалася на 27%, і зараз досягла показників у 4,1 мільярд гривень. І ця тенденція зберігається.

В вищих навчальних закладах напрям комп'ютерних наук шаленими темпами набирає популярність продовж кількох років. Якщо ще сім років тому цей напрям часто мав незаповнені бюджетні місця, то зараз конкурс у престижних вишах може сягати до 25 осіб на місце. Це говорить про привабливість професії серед молоді та високу конкуренцію.

ІТ компанії в Україні гарно фінансовані і співробітникам уділяється багато уваги, робітникам пропонують комфортні офісні приміщення, зони відпочинку, гнучкі графіки роботи, страхування і навіть спортзали.

Усі ці фактори, а в особливості гарні умови праці та дуже висока ЗП, призводять до того що значна кількість людей обирають роботу в ІТ індустрії, і йдуть до університетів для навчання чи за допомогою курсів

					ДП ІС-5119.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

покидають свою стару роботу і влаштовуються в ІТ компанії. Загальні факти про стан ІТ індустрії наводяться в [1].

ІТ галузь охоплює багато професій, і програмування це лише одна з них. Також є професії пов'язані з менеджментом, дизайном, маркетингом, аналітикою та іншими. Крім того, обравши професію програмування треба визначитись з мовою програмування яку треба вивчити і постійно відточувати свої знання і навички, бо програміст що розвивається повільніше ніж комп'ютерні технології може бути витісненим конкуренцією. Саме з метою допомоги у виборі шляху свого подальшого розвитку було створено систему аналізу ринку праці фахівців з інформаційних технологій (далі система аналізу).

Система аналізу дозволяє користувачу визначитись з наступними питаннями:

- чи потрібне для працевлаштування отримання вищої освіти;
- чи потрібно витратити час на збільшення рівню англійської мови;
- яка мова програмування буде популярною / високо оплачуваною в майбутньому і яку вивчити;

Відповіді на ці питання будуть даватися на основі статистичних даних, зібраних упродовж останніх десяти років, які були сформовані на основі соціального опитування користувачів сайту з пошуку вакансій jobs.dou.ua, результати якого знаходяться у вигляді файлів розширення csv на сайті github.com [2]. Також великий проміжок часу на протязі якого збиралися ці дані дозволяє нам спробувати передбачити поведінку ринку в недалекому майбутньому.

1.1.1 Опис процесу діяльності

Ціль користувача – отримати актуальну статистичну інформацію про показники середньої ЗП в галузі інформаційних технологій та прогноз зарплатні в майбутньому, для прийняття рішень щодо вибору напрямку для інвестування. Схема структурна послідовності системи аналізу наведена в графічному матеріалі.

1.1.2 Опис функціональної моделі

Опис функціональної моделі системи в умовах автоматизації у вигляді структурної схеми варіантів використання наведено в графічному матеріалі.

Обов'язкові до реалізації наступні функціональні вимоги:

- система повинна надати користувачу форму для заповнення і кнопку для відправлення запиту;
- система повинна повернути результат запиту у вигляді графіків.

Дуже бажані до реалізації наступні функціональні вимоги:

- Система має надавати можливість повернутися до вікна формування запиту для його редагування і повторного відправлення.

Бажані до реалізації наступні функціональні вимоги:

- Система має мати інтерфейс, що буде дозволяти доступ з мобільних девайсів чи персональних комп'ютерів.

1.2 Огляд наявних аналогів

З доступних в мережі Інтернет безкоштовних аналогів можна виділити статті аналітиків експертів на форумах та аналіз даних опитування на сайті jobs.dou.ua [3].

В порівнянні з цими статтями експертів наша система буде мати наступні переваги:

- завжди актуальна. Система збирає статистику на основі соціальних опитувань які проводяться щотижня, а стаття на форумі буде містити лише дані наявні на час її написання;
- наочне відображення інформації. Графіки наочно демонструють статистичні результати;
- математична екстраполяція. Майбутнє прогнозується математичними методами.

В порівнянні з аналізом даних від dou.ua [3], приклад якого зображено на рисунку 1.3, наша система буде мати наступні переваги:

- наявність графіків;
- інформація збирається з усіх опитувань одночасно;
- математичне прогнозування.

Зарплаты разработчиков —

декабрь 2018

Город:

Киев

Должность:

Junior QA engineer

Специализация:

Опыт: меньше года – 10+ лет

I квартиль

\$420

Медиана

\$540

III квартиль

\$750

Анкет: 133

Рисунок 1.3 – Аналог аналітичного віджету від dou.ua

Порівняємо з іншим віджетом: doustatistic.byethost7.com [4] зразок на рисунку 1.4.

Набор данных:

Май 2011

Декабрь 2011

Май 2012

Декабрь 2012

Май 2013

Декабрь 2013

Май 2014

Декабрь 2014

Май 2015

Декабрь 2015

Май 2016

Декабрь 2016

Июнь 2017

Декабрь 2017

Июнь 2018

Декабрь 2018

Образование:

Все

Высшее (8096)

Незаконченное высшее (1131)

Два высших (635)

Техникум / Колледж (208)

Среднее (170)

Кандидат (139)

Позиция:

Все

Software Engineer (2521)

Senior Software Engineer (1295)

Junior Software Engineer (1195)

QA engineer (788)

Project manager (477)

HR (410)

Junior QA engineer (405)

Technical Lead (381)

Senior QA engineer (334)

Designer (303)

Язык:

Все

JavaScript (1378)

Java (964)

C#/.NET (849)

PHP (758)

Python (293)

Swift (233)

C++ (226)

Ruby/Rails (163)

Kotlin (143)

Scala (82)

Город:

Все

Киев (4601)

Харьков (1601)

Львов (1241)

Днепр (748)

Одесса (550)

Удаленно (260)

Винница (231)

Запорожье (173)

Другой (161)

Пвано-Франковск (103)

Всего анкет: 10379

Учитывать только 3П ниже: 0

Показывать анкет: 50

Рассчитать

Добавить в сравнения

Среднее

Распределение

\$1984

\$80+

\$460+

\$840+

\$1220+

\$1600+

\$1990+

\$7992+

\$13994+

\$19996+

\$25998-\$32000

10379

936

1701

1443

946

823

4459

38

1

0

2

Рисунок1.4 – Віджет doustatistic.byethost7.com [4]

Переваги:

- аналіз усіх опитувань одночасно;
- менша кількість параметрів збільшують відносну вибірку анкет, які їм відповідні;
- графічне представлення;
- в віджеті не враховано несумісність деяких параметрів.

1.3 Постановка задачі**1.3.1 Призначення розробки**

Призначенням розробки є створення сервісу що буде допомагати людям приймати рішення щодо їх подальшого навчання, з метою майбутнього працевлаштування. Допомога полягає в наданні сервісу для перегляду обробленої статистичної інформації що стосується працевлаштування в ІТ індустрії, з використанням прогнозування трендів та графічного відображення інформації.

1.3.2 Цілі та задачі розробки

Мета дослідження – спростити користувачу пошук ІТ професії шляхом розробки комплексу задач аналізу ринку праці фахівців ІТ і інструменту візуалізації показників середньої ЗП з результатів опитувань працівників ІТ, доступних в мережі Інтернет.

Задачі розробки наступні:

- розробити графічний інтерфейс для введення запиту від користувача;
- розробити структуру БД для зберігання результатів опитувань, на основі яких робиться аналіз;

					ДП ІС-5119.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

- використати інструмент для побудови графіків що відображають зміну ЗП та популярності;
- надати можливість користувачу створювати складні запити;
- створити систему прогнозування показнику середньої ЗП фахівців ІТ.

Висновок до розділу

Проаналізована актуальність знаходження роботи в галузі ІТ індустрії на даний момент, описана цільова аудиторія користувачів системи аналізу. Розглянуті наявні аналоги аналізу статистичної інформації ринку праці галузі ІТ та зроблені порівняння. Визначені призначення, цілі та задачі розробки.

					ДП ІС-5119.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

У данному розділі описано структуру вхідних даних, яку система отримує від користувача. Введення даних відбувається через веб-форму яка є частиною програми. Форма описана в таблиці 2.1.

Таблиця 2.1 – Вхідні документи

Назва	Умове позначення	Розрядність
Форма запиту	requestForm	Об'єкт

Після заповнення форма розбивається на компоненти, які використовуються при фільтрації даних опитувань. Компоненти форми описані в таблиці 2.2.

Таблиця 2.2 – Вхідні дані

Назва	Умове позначення	Розрядність
Місто	city	Текст
Посада	position	Текст
Мова програмування	language	Текст
QA спеціалізація	specialization	Текст
Рівень знання англійської мови	englishLevel	Текст
Рівень освіти	educationLevel	Текст
Мінімальний досвід роботи	minimalExperience	Число від 1 до 10
Максимальний досвід роботи	maximalExperience	Число від 1 до 10

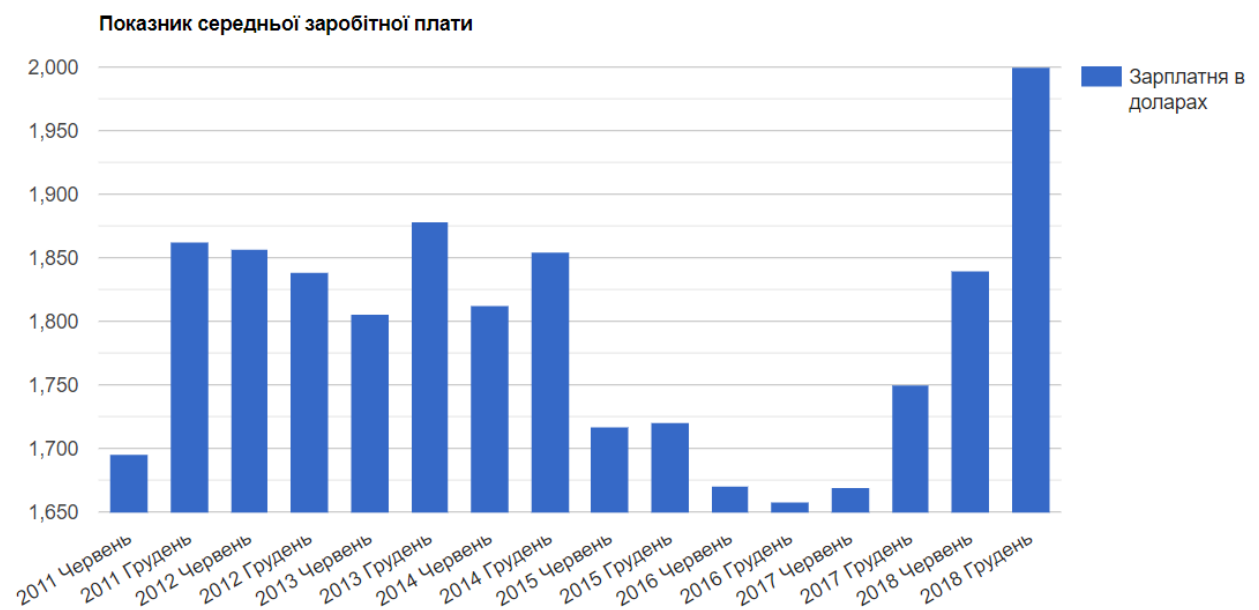
2.2 Вихідні дані

Вихідні дані програма повертає на сторінку, в якій користувач заповнював форму запиту. Вихідними даними є два набори точок для побудови графіків а також самі графіки. Вихідні дані зображені в таблиці 2.3.

Таблиця 2.3 – Вихідні дані

Назва	Умове позначення	Розрядність
Точка	point	[дата,значення]
Набір точок для графіку тенденції ЗП	salaryGraph	Впорядкована за часом колекція точок, розміром 16.
Набір точок для графіку тенденції популярності	popularityGraph	Впорядкована за часом колекція точок, розміром 16.

Для відображення вихідних даних використовуються відповідні



графіки. Приклад графіків зображено на рисунках 2.1 та 2.2.

Рисунок 2.1 – Приклад вигляду вихідного графіку для зарплатні

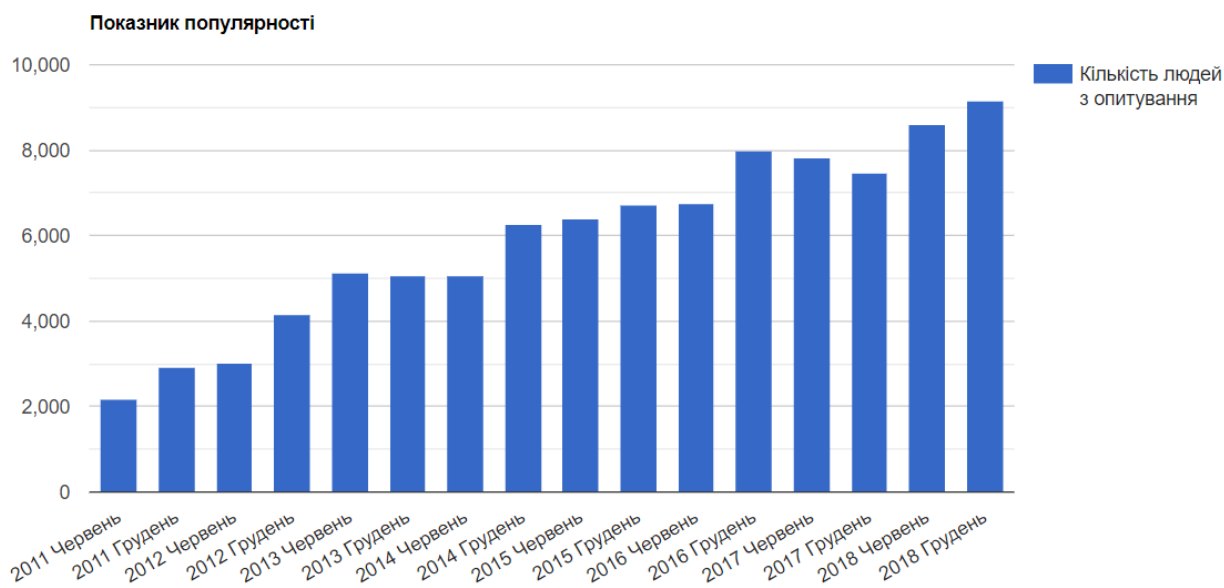


Рисунок 2.2 – Приклад вигляду вихідного графіку для популярності

2.3 Опис структури бази даних

Результати опитувань зберігатимуться в БД у вигляді єдиної таблиці, структура якої зображена на рисунку 2.3. Ця Таблиця денормалізована [5], це потрібно для пришвидшення операцій типу SELECT, які будуть єдиним типом операцій застосованим до цієї таблиці в процесі виконання програми. Також це полегшить перенесення даних з файлу формату csv до таблиці БД. Дані про результати опитувань були взяті з сайту github.com [2].

Діаграма відношень сутностей що описує структуру БД для зберігання результатів опитувань та структура таблиці для зберігання результатів опитувань наведені в графічному матеріалі.

Також в програмі використовується об’єкт, який зберігає дані про всі варіанти параметрів опитування, що використовуються для формування форми запиту користувача. Для того щоб при кожному запуску програми не перевизначити цей об’єкт, він також буде зберігатись в БД у вигляді схеми зірки [6]. Діаграма відношень сутностей що описує структуру БД для

зберігання варіантів параметрів опитувань та структура таблиці для зберігання результатів опитувань наведені в графічному матеріалі.

Результату запиту користувача буде відповідати набір точок. Вигляд структури для зберігання точок у вигляді програмного коду можна знайти в додатку А, разом з кодом програми. Шукати потрібно в класі з назвою Point.

Графік задається набором точок і відображається за допомогою утиліти googleCharts, якій передається інформація для відображення у вигляді масиву пар інформації наступного вигляду. Для зручності розуміння, назва дати опитування була перекладена українською мовою. Об'єкт що зберігає точки після трансформації несе назву GoogleAPIPoint. Вигляд його структури також потрібно шукати в відповідному класі – GoogleAPIPoint.

Зберігання точок в БД недоцільно, бо кожна комбінація параметрів запиту користувача буде призводити до утворення різних точок.

Висновок до розділу

Визначено структуру вхідних сигналів і даних та вихідних сигналів і документів. Спроектована структура БД для зберігання інформації необхідної для створення та обробки запиту користувача аналітичної системи. Обґрунтований вибір форм таблиць БД.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Наявні дані соціальних опитувань людей що працюють в галузі ІТ за 2011 – 2018 роки. Використовуючи ці дані необхідно виконати прогноз тенденцій для наступних років. При використанні нейронної мережі, постає задача знаходження факторів що будуть впливати на результат прогнозування та включення їх в мережу, з метою збільшення точності прогнозування та спрощення процесу навчання мережі. Після дослідження процесу використання нейронної мережі для передбачень стану зарплатні, зробити висновки щодо можливості її використання в даній програмі.

3.2 Математична постановка задачі

Під виділенням тенденцій мається на увазі знайти табір точок, в яких координата X позначає час опитування а координата Y позначає середню ЗП. Такий набір точок має назву часовий ряд. Часовий ряд містить результати спостереження за процесом на деякому інтервалі часу, званому ділянкою спостереження. Відрізок часу від останнього спостереження до того моменту, для якого нам необхідно отримати прогноз, називається ділянкою упередження. Приклад часового ряду зображено на рисунку 3.1. Задача: скласти часовий ряд для ділянки упередження.

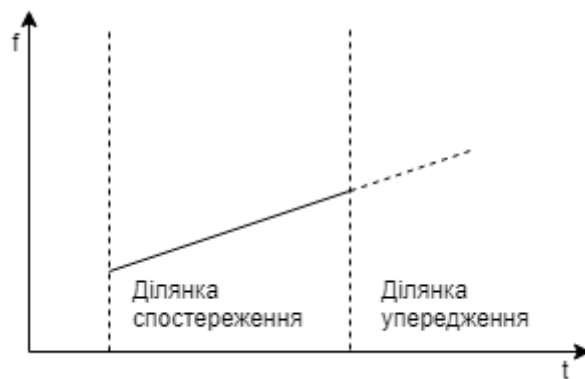


Рисунок 3.1 – Приклад часового ряду

3.3 Обґрунтування методу розв'язання

Обирати метод вирішення цієї задачі будемо з двох кандидатів: метод поліноміальної екстраполяції та прогнозування за допомогою нейронних мереж. Порівняємо ці методи.

Приклад поліноміальної екстраполяції зображено на рисунку 3.2

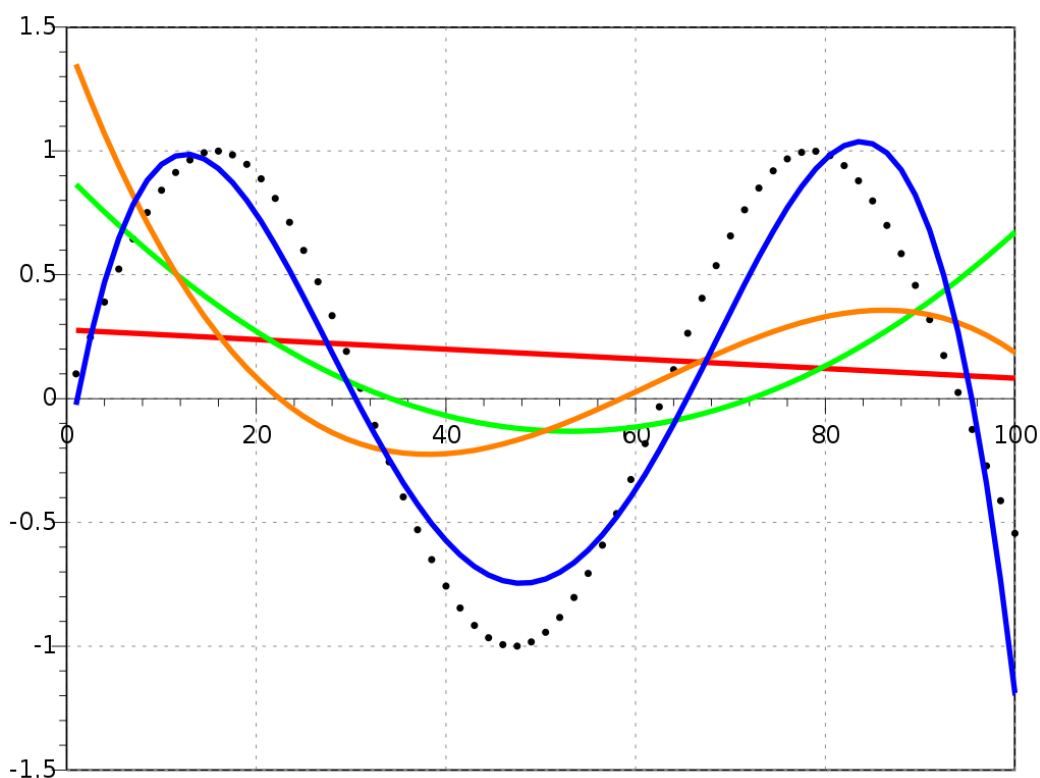


Рисунок 3.2 – Приклад поліноміальної екстраполяції для функції синуса

Цей метод використовують для отримання функції що ближче всіх проходить по заданих вхідних точках. Основний недолік цього методу полягає в непередбаченій поведінці функції поза межами ділянки спостереження, і чим більший порядок поліному, тим більші коливання кривої, як видно з рисунку 3.2, де червоним кольором позначено поліноміальну функцію першого порядку, а зеленим помаранчевим та синім позначені функції другого третього і четвертого порядків відповідно.

Альтернативним варіантом прогнозування виступають штучні нейронні мережі[7, 8]. Зокрема в [9] розглядається питання використання нейронних мереж у фінансовій сфері.

З плином часу, інформація що потребує аналізу стає все складнішою та більш об'ємною, і швидкість зміни інформації тер зростає, що робить задачу аналізу все більш вимогливою до часу виконання. Обробка такої інформації людським інтелектом стає неможливою, а використання традиційних математичних засобів стає занадто вимогливим до ресурсів процесом. Тому на заміну традиційним методам для аналізу даних приходять нейронні мережі. Зараз нейронні мережі надають розробникам потужні можливості для досягнення цілей [10].

Нейронна мережа була створена як спроба математично змодельовати процеси що проходять в біологічних нейронах що об'єднані в одну функціонуючу систему. Хоч нейромережа і являється значно спрощеною системою тваринного мозку, вона досить успішно використовується у вирішенні різноманітних задач таких як прогнозування чи розпізнавання образів [11, 12]. В майбутньому, з розвитком технологій, нейронні мережі навіть зможуть бути використані в створенні розумних мислячих комп'ютерів [13].

Але навіть зараз, коли нейронні мережі тільки починають користуватися популярністю, деякі компанії вже їх використовують для ігор

на біржі. Завдяки використанню нейронної мережі, фірма Richard Borst змогла збільшити свій фінансовий оборот на 6%.

Нейронні мережі, в порівнянні з традиційними математичними методами мають набір переваг, це:

- швидкість роботи;
- здатність давати результат, спираючись на неповну чи зашумлену інформацію;
- здатність налаштування мережі для отримання результатів максимальної точності в кожному конкретному випадку.

Виходячи з наведених переваг, для використання в програмі використовуємо нейронну мережу. Для моделювання мережі було обрано засіб nntool пакету MATLAB, бо навички роботи з ним були отримані в ході навчання та через те що цей засіб володіє потужними засоби візуалізації та навчання нейронних мереж [14].

3.4 Опис методу розв'язання

Для розв'язання будемо використовувати мережу прямого поширення з алгоритмом зворотного поширення похибки бо такі мережі часто використовують для моделювання часових рядів.

Вхідні дані

Результати опитувань фахівців галузі ІТ за 2011 – 2018р в форматі csv на сайті github.com [2].

Постановка задачі.

Дано. Часовий ряд рівня ЗП ІТ фахівців за півріччя (2011-2018р).

Зробити. Прогноз рівня ЗП на наступне півріччя за допомогою штучної нейронної мережі.

Підзадачі, які необхідно вирішити:

- вибрати ознаки вхідного вектора навчальної вибірки;
- вибрати тип і структуру штучної нейронної мережі;
- навчити нейронну мережу і емпірично підібрати параметри навчання.

Особливість для моделювання ЗП в тому, що за її збільшенням, дуже часто йде зменшення. Тому для нейронної мережі вибрані дві вхідні ознаки x_1 – рівень ЗП і x_2 – ознака зміни, якщо на попередньому кроці було зменшення, то $x_2=1$, якщо збільшення, $x_2=2$.

Таким чином загальна вибірка з даних сайту за 2011- 2018 роки виглядає наступним чином (для простоти моделювання півріччя позначені числами від 1 до 16:

- $x_1 = 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16$;
- $x_2 = 2 \ 2 \ 1 \ 1 \ 1 \ 2 \ 1 \ 2 \ 1 \ 2 \ 1 \ 1 \ 2 \ 2 \ 2 \ 2$;
- $y = 1695 \ 1863 \ 1857 \ 1839 \ 1806 \ 1878 \ 1813 \ 1855 \ 1717 \ 1721 \ 1670 \ 1658$
 $1669 \ 1750 \ 1840 \ 2000$.

Із загальної вибірки сформуємо навчальну вибірку без останніх даних для перевірки точності прогнозу після навчання. Текст програми MATLAB з навчанням нейронної мережі та її конфігурацією наведено в додатку А, разом з програмним кодом.

Графік що відображає вхідні дані зображено на рисунку 3.3.

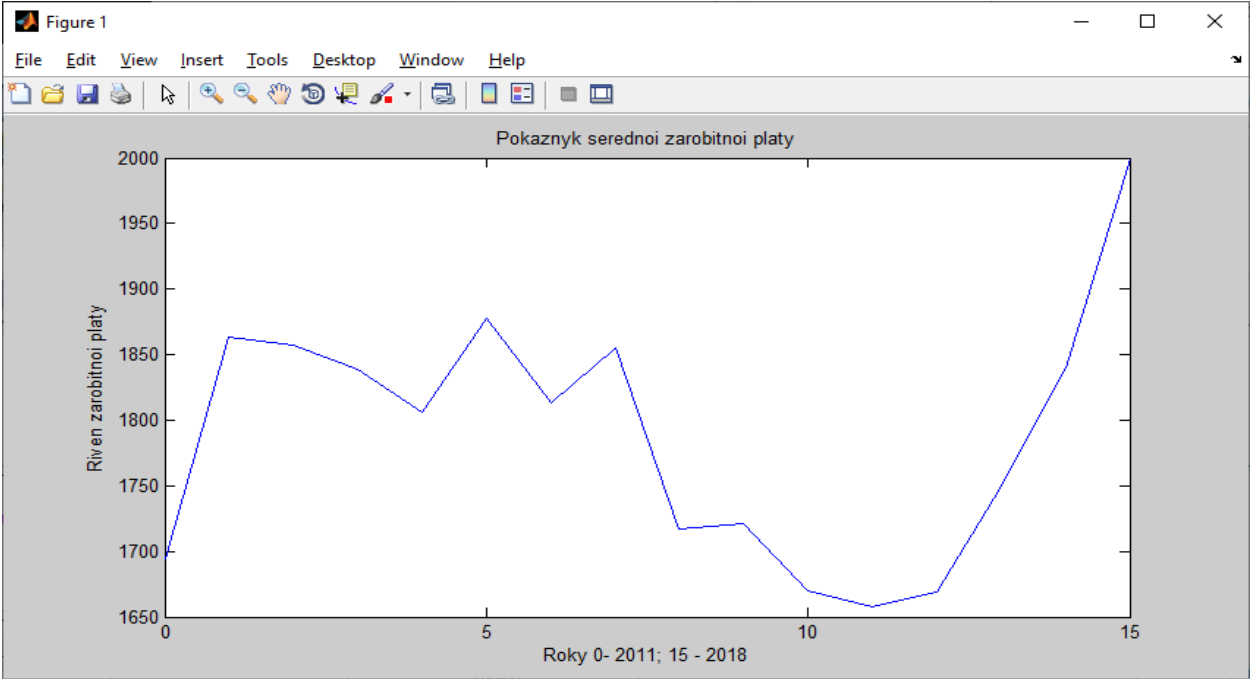


Рисунок 3.3 – Візуалізація вхідних даних

Структура нейронної мережі зображена на рисунку 3.4. Мережа складається з трьох шарів. Перший шар містить дванадцять нейронів, другий та третій по одному. В першому шарі використовується сигмоїдна порогова функція, в інших – лінійна. Мережа з даною конфігурацією була навчена за 9 епох. Навчання було зупинене завчасно через досягнення необхідної точності результату.

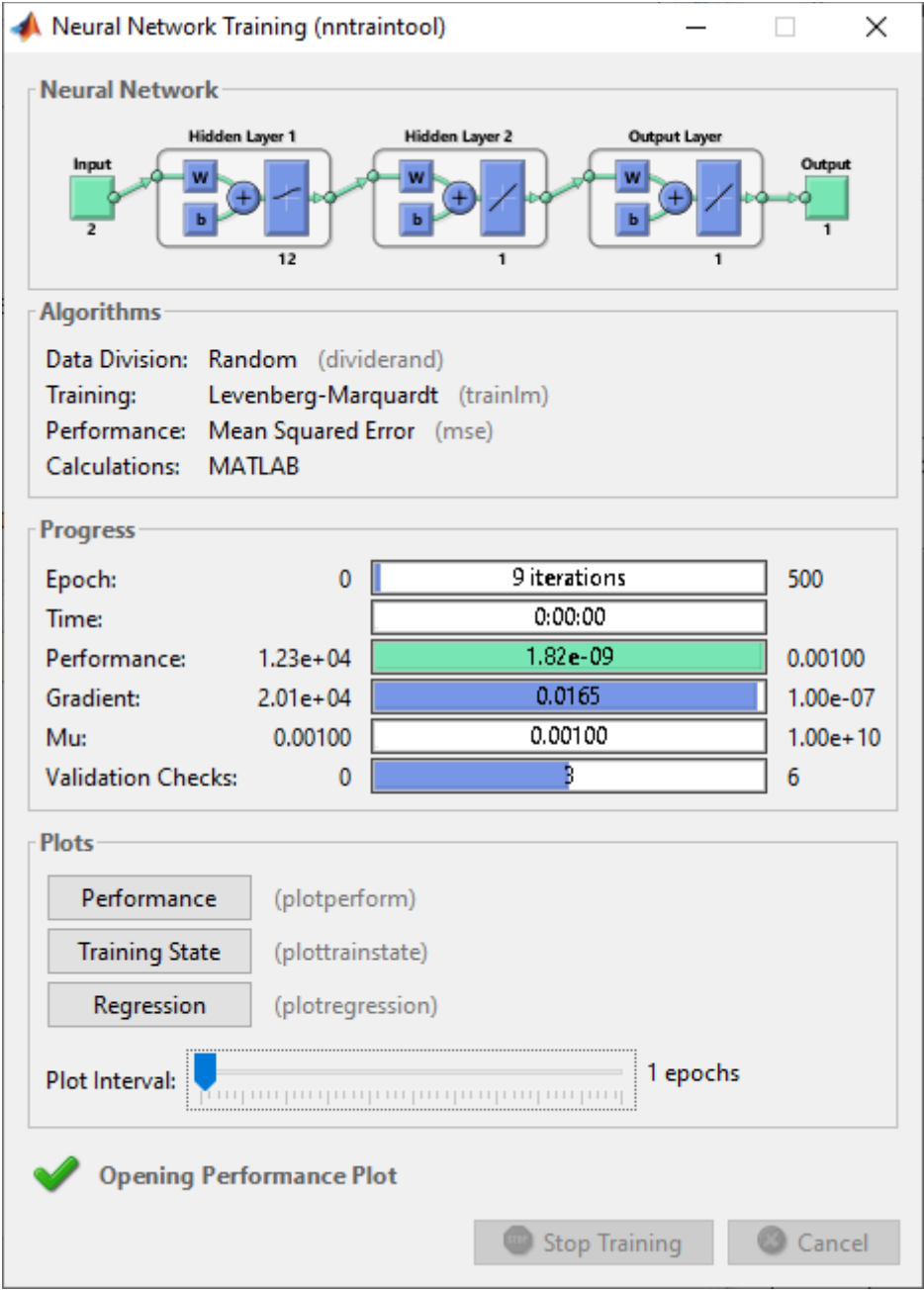


Рисунок 3.4 – Структура нейронної мережі

На рисунку 3.5 можна побачити візуалізацію процесу тренування мережі впродовж дев'яти епох.

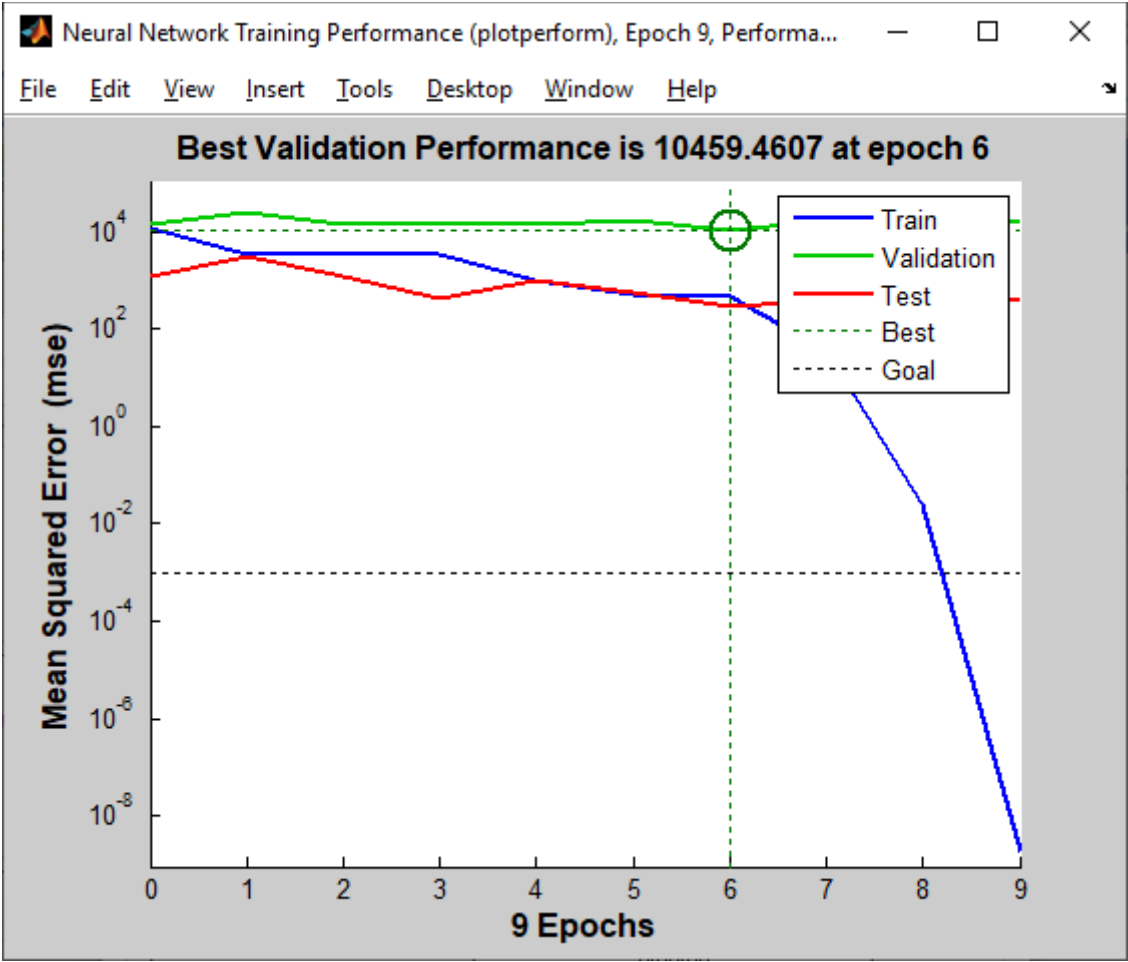


Рисунок 3.5 – Результат тренування нейронної мережі

Результат моделювання прогнозування згідно з програмою

$c = 1.0e+03 *$
Columns 1 through 6
1.6987 1.7198 1.8752 1.8443 1.8153 1.8808
Columns 7 through 12
1.7729 1.8247 1.7448 1.7241 1.7012 1.6705
Columns 13 through 15
1.6763 1.7323 1.8382
 $c2 = 1.9210e+0374.$

Прогноз програми на друге півріччя 2018-го року – 1921, в дійсності він був 2000. Відхилення від реального результату в даному випадку становить 4%.

Нейронна мережа навчалась декілька разів. Остаточний варіант структури і значень коефіцієнтів мережі був обраний для випадку, коли для останніх трьох значень навчальної вибірки досягалась достатня точність на виході нейронної мережі.

Подібні штучні нейронні мережі (на таких засадах) створюються для інших показників сайту, в тому числі для популярності. Таким чином користувач системи може отримати графік зміни середньої ЗП і прогноз на наступний термін.

Результат прогнозування на наступне півріччя

$$c = 1.0e+03 *$$

Columns 1 through 6

1.6713 1.8510 1.8270 1.7954 1.7884 1.8589

Columns 7 through 12

1.7789 1.7872 1.7087 1.6460 1.6617 1.6654

Columns 13 through 16

1.6773 1.7372 1.8400 1.9909

$$c2 = 2.0920e+03$$

Прогноз програми на перше півріччя 2019-го року – 2092 програма прогнозує збільшення зарплатні на 92\$ чи на 4,6% в порівнянні з попереднім півріччям.

У майбутньому система буде ускладнюватись шляхом додавання нових змінних: рейтингу міст, рейтингу мов програмування, рейтингу професійного досвіду. Система, що розробляється це передбачає і це описано в інших розділах.

					ДП ІС-5119.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

Висновок до розділу

Розглянуті різні методи вирішення задачі, порівняні їх переваги та недоліки. Для обраного методу детально розписано принцип роботи та наведено наочний приклад роботи програми. На практиці доведено можливість використання нейронної мережі для прогнозування ЗП, що працює зі статистичними даними отриманими в результаті роботи нашого програмного продукту. Можливе подальше виявлення нових факторів впливу на значення середньої ЗП та включення їх до нейронної мережі, з метою збільшення точності прогнозування.

					ДП ІС-5119.1181-с.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

У таблиці 4.1 наведено основні засоби що використовувались при розробці.

Таблиця 4.1 – Засоби розробки

Вид засобу	Назва засобу
Операційна система	Windows 10
Мови програмування	Java, JavaScript, html, MATHLAB, SQL
Середовища розробки	IntelliJ Idea, MATHLAB, MySQL Workbench
Додаткові засоби	google charts, Spring MVC, Thymeleaf, Maven, JPA
Засоби проектування	draw.io

В якості ОС для розробки була обрана Windows 10, через те що вона більш популярна зручна та стабільна ніж свої основні конкуренти Linux та MacOS. Але використання мови Java та застосування шаблону проектування MVC (“Модель – Представлення - Контролер”) [15] для розділення графічного інтерфейсу користувача від керуючої логіки да даних програми, та використання мов html та JavaScript для відображення інтерфейсу користувача, дозволяють реалізувати кросплатформність програми. Тобто програма може бути запущена на більшості ОС, таких як Linux, Windows, MacOS. Єдиною вимогою для запуску програми залишається присутність

веб-браузеру для відображення інтерфейсу користувача. Та встановлена програма MATLAB для використання нейронної мережі для прогнозування.

Програмний продукт був розроблений використовуючи фреймворк Spring MVC який використовують для створення сайтів та веб-застосунків за шаблоном проектування MVC. Обрано було саме Spring MVC, бо зараз Spring не має здатних конкурентів в області модернізації мови Java.

Паттерн програмування MVC передбачає розділення програми на три логічні частини: Представлення, Модель і Контролер.

Представлення – відповідає за відображення графічного інтерфейсу користувача. В даному випадку вона представлена html – сторінкою.

Модель – необхідна для опису об'єктів що зберігають дані з якими працює програма.

Контролер – відповідає за всі логічні операції з даними описаними в моделі та за передачу даних від інтерфейсу користувача до програми і навпаки.

Графічне відображення паттерну MVC зображено на рисунку 4.1.

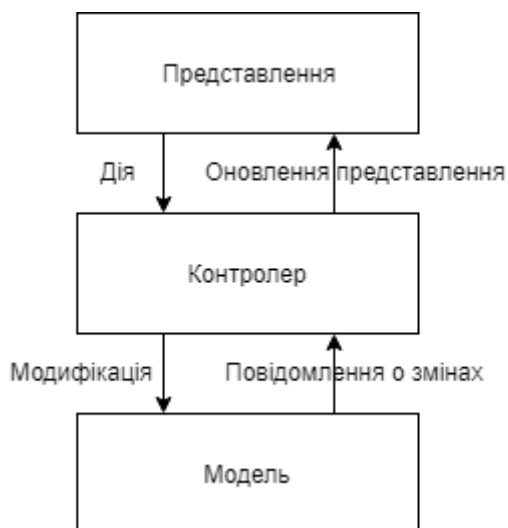


Рисунок 4.1 – Схема паттерну MVC

Java, як мова програмування була обрана за зручність розробки, кросплатформність, що забезпечується її особливістю компіляції з використанням віртуальної машини JVM [16], та наявністю багатьох бібліотек створених для неї програмістами впродовж років її існування. Для моделювання нейронної мережі була обрана мова та середовище MATLAB, через те що інструменти тренування нейронних мереж в MATLAB значно перевищують по зручності та потужності інструменти Java чи інших мов програмування високого рівня.

Середовищем розробки для програми написаній на Java було обрано IntelliJ Idea, бо воно безкоштовне і значно полегшує процес розробки за рахунок вбудованого компілятора, пошуку програмних помилок та інших корисних інструментів розробки вигляд середовища зображений на рисунку 4.2. Середовище MySQL Workbench використовувалось для візуалізації БД програми та для перенесення даних опитувань до локальної БД середовище зображено на рисунку 4.3.

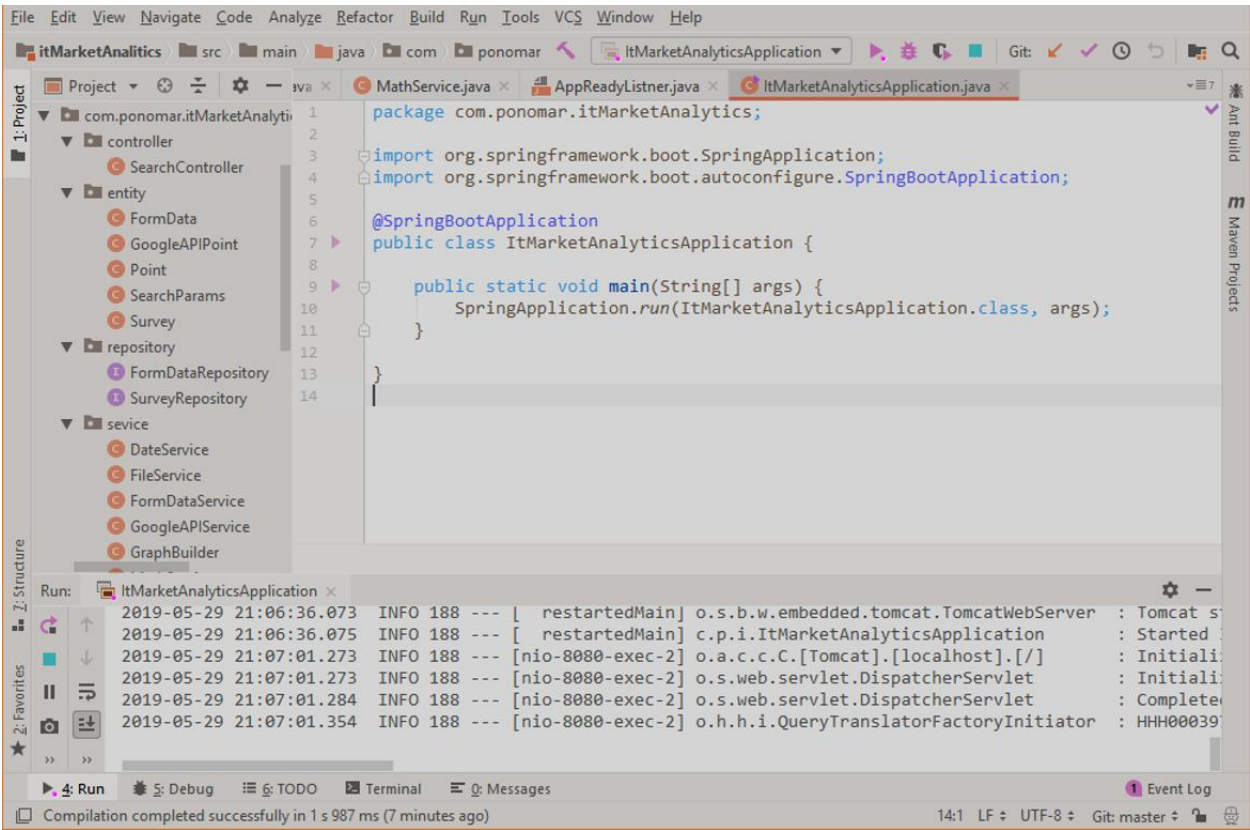


Рисунок 4.2 – Середовище програмування IntelliJ Idea

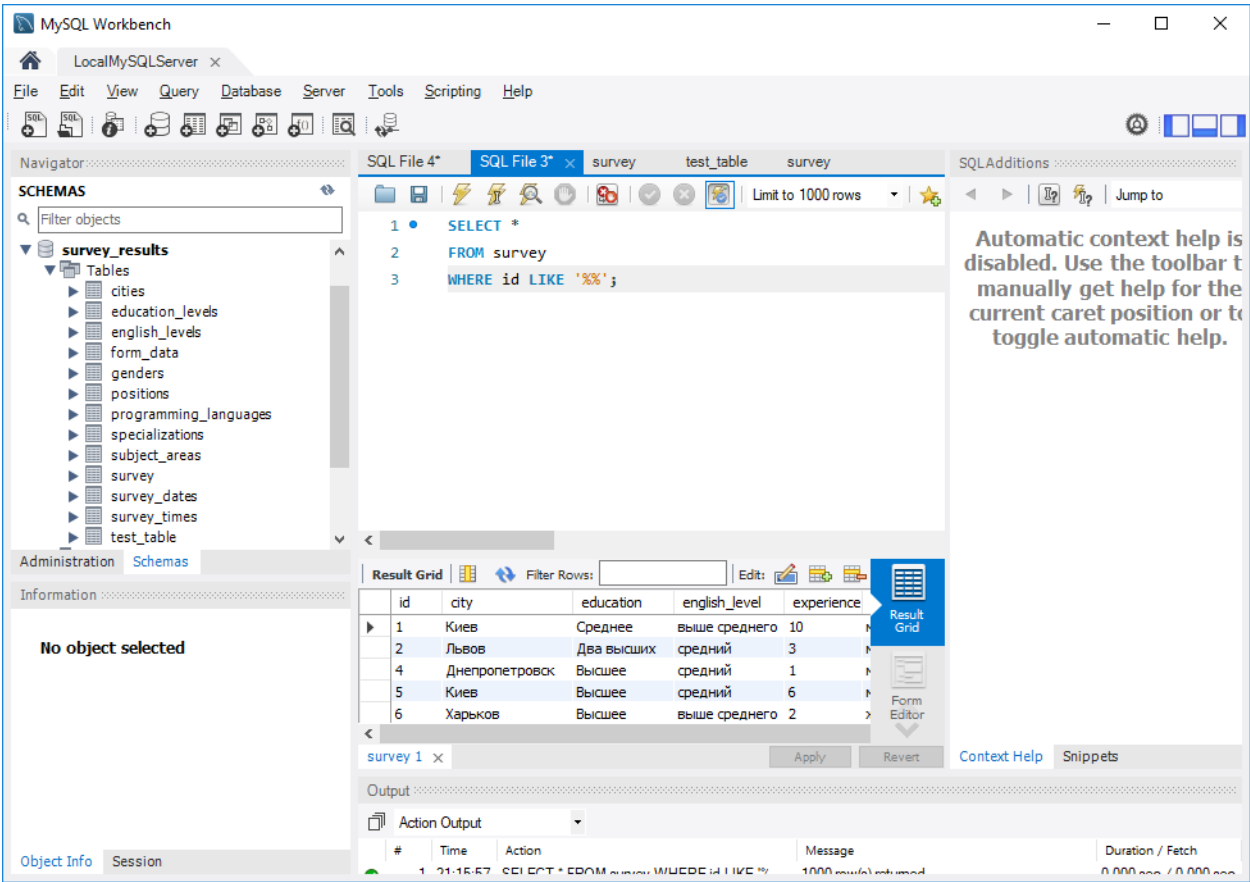


Рисунок 4.3 – Середовище роботи з базами даних MySQL Workbench

З додаткових засобів було використано Google Charts API [17], який дозволяє будувати інтерактивні графіки. Приклад вигляду графіку зображено на рисунку 4.4.

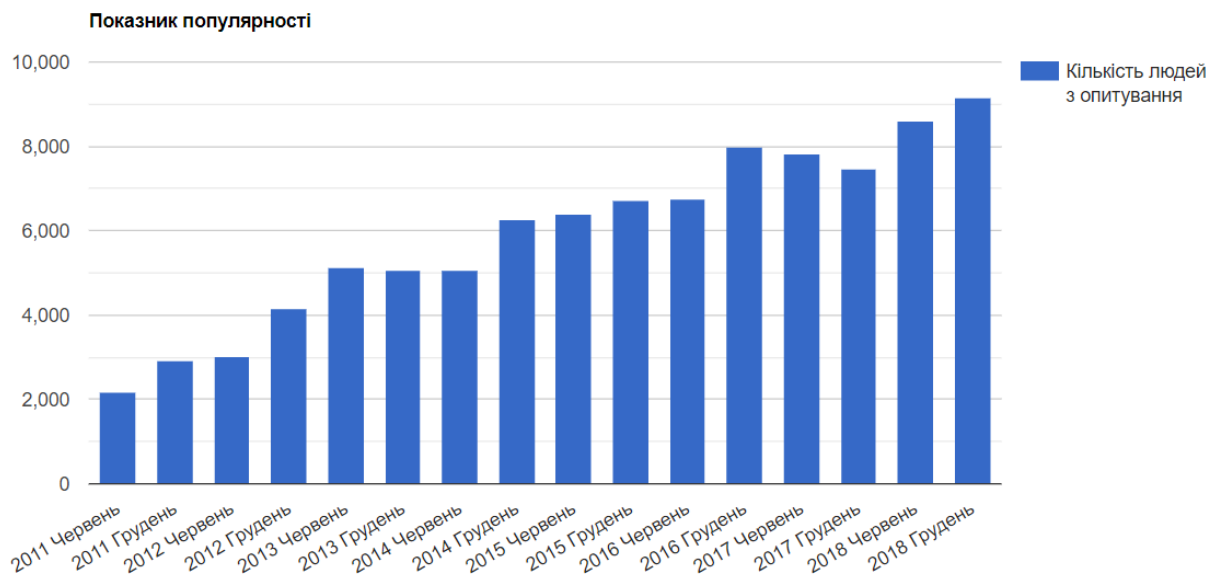


Рисунок 4.4 – Приклад графіку

Spring MVC – фреймворк для створення програм по шаблону проектування MVC [15].

Thymeleaf – механізм створення шаблонів html сторінок, допомагає при передачі параметрів від контролера до представлення і навпаки [18].

Maven – використовується для імпорту сторонніх бібліотек до проекту та його зборки.

JPA – інструмент, що дозволяє працювати з об'єктами БД як з Java - об'єктами, абстрагує зв'язок з базою даних і бере на себе задачу формування SQL запитів до БД [19].

Для проектування було обрано онлайн-ресурс draw.io. Це дуже гнучкий інструмент для створення графічної документації найрізноманітніших

направлень. Він має зручну систему збереження результатів роботи на гугл диску, зі здатністю подальшого редагування та простий, інтуїтивно зрозумілий інтерфейс. Приклад використання draw.io зображено на рисунку 4.5.

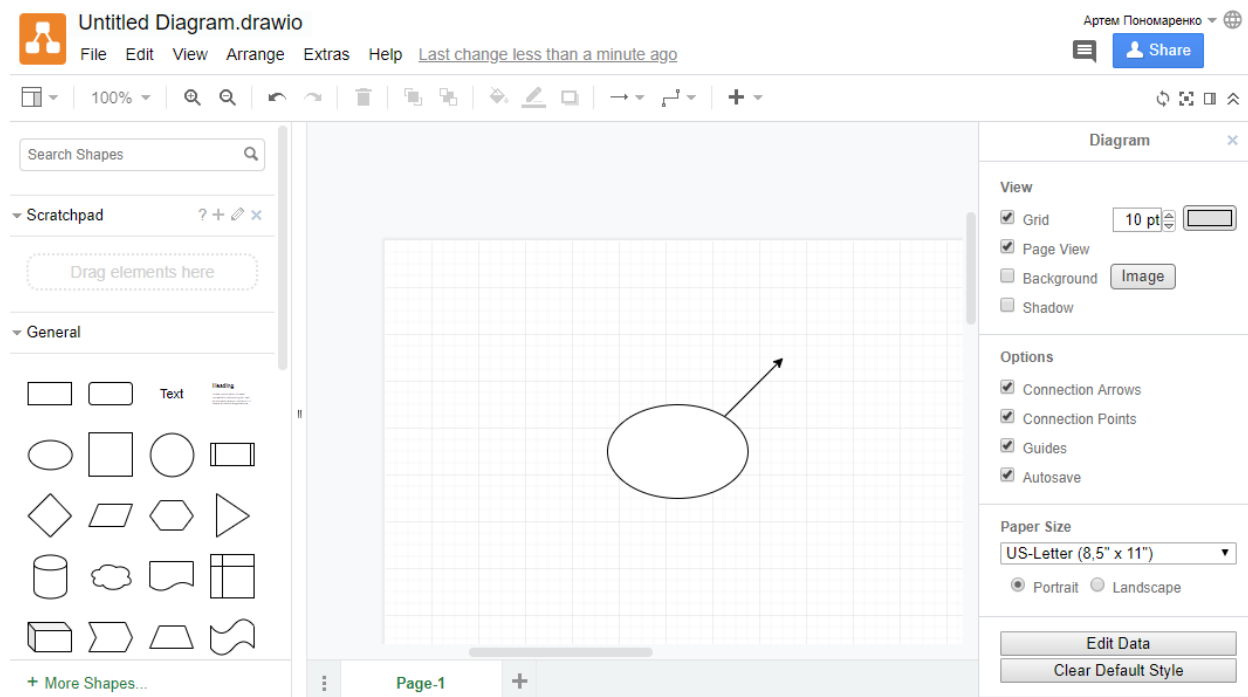


Рисунок 4.5 – Інтерфейс програми для створення діаграм

4.2 Вимоги до технічного забезпечення

Програмний продукт працює через веб-застосунок, тому єдиними вимогами до технічних засобів являються наявність одного з веб-браузерів, зазначеного в умовах експлуатації та наявність на пристрою користувача доступу до мережі Інтернет.

Зазначу технічні засоби необхідні для встановлення веб-браузеру Google Chrome в таблиці 4.2

Таблиця 4.2 – Системні вимоги

Вимога	Вимоги к Windows	Вимоги к Mac	Вимоги к Linux
Операційна система	Windows XP з пакетом оновлень 2 + Windows Vista Windows 7 Windows 8 Windows 10	Mac OS X 10.6 чи пізнішої версії	Ubuntu 10.04 + Debian 6 + OpenSuSE 11.3 + Fedora Linux 14
Процесор	Intel Pentium 4 / Athlon 64 чи пізнішої версії з підтримкою SSE2		
Вільне місце на диску	350 Мб		
Оперативна пам'ять	512 Мб		

Також для функціонування системи на персональних комп'ютерах необхідні наступні прилади периферії:

- монітор;
- клавіатура;
- тачпад чи миша.

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма класів

Структурна схема класів програмного забезпечення наведена в графічних матеріалах. Далі наведено опис функціонування кожного класу задіяного в системі.

SearchController – єдиний клас контроллер в проекті. Забезпечує передачу даних від представлення до програми і навпаки.

FormDataService – Забезпечує зв'язок з репозиторієм, відповідає за доступ до об'єкту FormData, який зберігає інформацію необхідну для відображення веб-форми.

FormRepository – забезпечує зв'язок з таблицею в БД, що зберігає об'єкт FormData.

FormData – описує структуру об'єкта FormData та вигляд таблиці в БД для його зберігання.

SurveyService – Забезпечує зв'язок з репозиторієм, відповідає за доступ до об'єктів Survey, які зберігають інформацію що відповідає результатам опитувань. Також відповідає за фільтрацію даних типу Survey відповідно запитам користувача.

SurveyRepository – забезпечує зв'язок з таблицею в БД, що зберігає об'єкти Survey.

Survey – описує структуру об'єкта Survey та вигляд таблиці в БД для його зберігання.

SurveySpecifications – описує методи для формування динамічних запитів до БД, за допомогою яких відбувається фільтрація даних у класі SurveyRepository.

SearchParams – клас-обгортка, в якому параметри запиту користувача передаються від класу SearchController до класу SurveyService для виконання фільтрації даних за цими параметрами.

GraphBuilderService – основний клас типу сервіс, відповідає за координацію інших сервісів для виконання однієї цілі: перетворити запит користувача в набір точок який буде використовуватися для відображення на графіках та побудови даних для аналізу за допомогою нейросітки.

Point – набір таких точок повертається в результаті виконання функції класу GraphBuilderService.

MathService – допомагає у виконанні базових математичних операцій.

DateService – допомагає в різних операціях пов'язаних с датами.

GoogleAPIService – робить фінальну трансформацію наборів точок типу Point в набір точок типу GoogleAPIPoint для передачі їх на представлення і відображення їх на графіку.

GoogleAPIPoint – Об'єкт формату гідного для відображення використовуючи GoogleGraphs.

4.3.2 Діаграма послідовності

Структурна схема послідовності наведена в графічних матеріалах. Детальний опис послідовності описано далі.

Користувач відкриває сторінку з веб інтерфейсом, заповнює представлену форму та тисне кнопку «відправити запит». Контролер зчитує дані з форми і пересилає їх моделі командою `getPoints`, очікуючи в результаті обробки отримати точки для побудови графіків. Модель, використовуючи параметри користувача для фільтрації даних, дістає зі сховища даних результати опитувань що відповідають запиту (`getServeysWithParams`) і перетворює інформацію в точки для графіків(`getAvarageSalary`). Набір точок повертається до контролера, потім пересилається назад в модель для форматування(`toAPIPoints`). Далі ці дані пересилаються з моделі в контролер, з контролеру в модель де оброблюються мовою JavaScript для відображення графіків користувачеві.

4.3.3 Діаграма компонентів

Програма складається з трьох компонентів: представлення, контролера і моделі.

В представлення(view) входить представлення веб-сторінки застосунку описане в файлі index.html.

В контролер входить клас SearchController, який керує передачею даних від сторінки до моделі.

В модель входять сховища даних для результатів опитувань та для інформації, необхідної для побудови представлення а також всі інші класи типу сервіс.

4.3.4 Специфікація функцій

Клас SearchController – єдиний клас контролер в проекті. Забезпечує передачу даних від представлення до програми і навпаки. Специфікація в таблиці 4.3.

Таблиця 4.3 – Опис функцій класу SearchController

Назва функції	Призначення функції
showIndex	Відобразити початкову сторінку
showResponse	Відобразити початкову сторінку разом з графіками

Клас FormDataService – Забезпечує зв'язок з репозиторієм, відповідає за доступ до об'єкту FormData, який зберігає інформацію необхідну для відображення веб-форми. Специфікація в таблиці 4.4.

Таблиця 4.4 – Опис функцій класу FormDataService

Назва функції	Призначення функції
refreshFormData	Викликається, якщо потрібно оновити об'єкт FormData після внесення змін в БД об'єкту Survey
getFormData	Звертається до репозиторію і повертає об'єкт FormData

Клас SurveyService – Забезпечує зв'язок з репозиторієм, відповідає за доступ до об'єктів Survey, які зберігають інформацію що відповідає результатам опитувань. Також відповідає за фільтрацію даних типу Survey відповідно запитам користувача. Специфікація в таблиці 4.5.

Таблиця 4.5 – Опис функцій класу SurveyService

Назва функції	Призначення функції
getAllSurveys	Повертає всі дані опитувань
getSurveysWithParams	Повертає лише ті дані опитувань, що відповідають вхідним параметрам

Клас SurveySpecifications – описує методи для формування динамічних запитів до БД, за допомогою яких відбувається фільтрація даних у класі SurveyRepository. Специфікація в таблиці 4.6.

Таблиця 4.6 – Опис функцій класу SurveySpecifications

Назва функції	Призначення функції
withCity	Створення шаблону для пошуку опитуваних лаше з вказаним містом роботи

Продовження таблиці 4.6

withEducation	Створення шаблону для пошуку опитуваних лаше з вказаним рівнем освіти
withEnglishLevel	Створення шаблону для пошуку опитуваних лаше з вказаним рівнем знання англійської мови
withExperienceLessThan	Створення шаблону для пошуку опитуваних лаше з досвідом роботи менше вказаного
withExperienceMoreThan	Створення шаблону для пошуку опитуваних лаше з досвідом роботи більше вказаного
withPosition	Створення шаблону для пошуку опитуваних лаше з вказаною посадою
withProgrammingLanguage	Створення шаблону для пошуку опитуваних лаше з вказаною мовою програмування
withSpecialization	Створення шаблону для пошуку опитуваних лаше з вказаною спеціалізацію в області QA

Клас GraphBuilderService – основний клас типу сервіс, відповідає за координацію інших сервісів для виконання однієї цілі: перетворити запит користувача в набір точок який буде використовуватися для відображення на графіках та побудови даних для аналізу за допомогою нейро мережі. Специфікація в таблиці 4.7.

Таблиця 4.7 – Опис функцій класу GraphBuilderService

Назва функції	Призначення функції
getPoints	Збирає дані відповідно запиту з SurveyService. Знаходить значення популярності, середньої зарплатні за кожен період опитування і записує ці дані в набір точок, впорядкованих за часом проведення опитування. Повертає в контролер набір точок для побудови графіку

Клас MathSetvice – допомагає у виконанні базових математичних операцій. Специфікація в таблиці 4.8.

Таблиця 4.8– Опис функцій класу MathSetvice

Назва функції	Призначення функції
getAverage	Знаходить середнє арифметичне, ділячи загальну суму на кількість опитованих. Запобігає діленню на нуль

Клас DateService – допомагає в різних операціях пов'язаних с датами. Специфікація в таблиці 4.9.

Таблиця 4.9 – Опис функцій класу MathSetvice

Назва функції	Призначення функції
dateToString	Повертає текстове відображення даних формату Date.

Продовження таблиці 4.9

transformDates	Перекладає назву місяців на Українську мову, прибирає день опитування
----------------	---

Висновок до розділу

У розділі програмне та технічне забезпечення було описано засоби, що застосовані для розробки програмного продукту та обґрунтовано їх вибір. Поставлені вимоги до технічного забезпечення. Наведено інформацію щодо архітектури розробленого програмного забезпечення, описано діаграми, що її пояснюють. Приведено детальний перелік функцій класів, які складають систему. Спроековано веб-форму для введення запитів користувача та відображення графіків. Налагоджено процес перетворення даних опитувань на точки для побудови графіків.

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

При відкритті веб-застосунку користувач бачить веб-форму яка зображена на рисунку 5.1

Місто

вся Україна

Посада

будь-яка

Мова програмування

будь-яка

Рівень англійського

будь-який

Рівень освіти

будь-який

Досвід більший ніж:

Менше року

Досвід менший ніж:

Більше десяти років

Відправити запит

Рисунок 5.1 – Стартова сторінка застосунку

Далі користувач обирає зі спадного списку параметри для пошуку. Приклад заповненої форми відображено на рисунку 5.2.

Місто

вся Україна

Посада

Software Engineer

Мова програмування

Java

Рівень англійського

будь-який

Рівень освіти

Высшее

Досвід більший ніж:

Менше року

Досвід менший ніж:

Більше десяти років

Відправити запит

Рисунок 5.2 – Приклад заповненої форми

Після того як користувач обрав необхідні параметри і натиснув кнопку «відправити запит», програма обробляє його запит і виводить на сторінку графік зміни зарплатні, зображений на рисунку 5.3, та графік зміни популярності, зображений на рисунку 5.4. разом з формою, яку можливо використати для подальшого відправлення запитів.

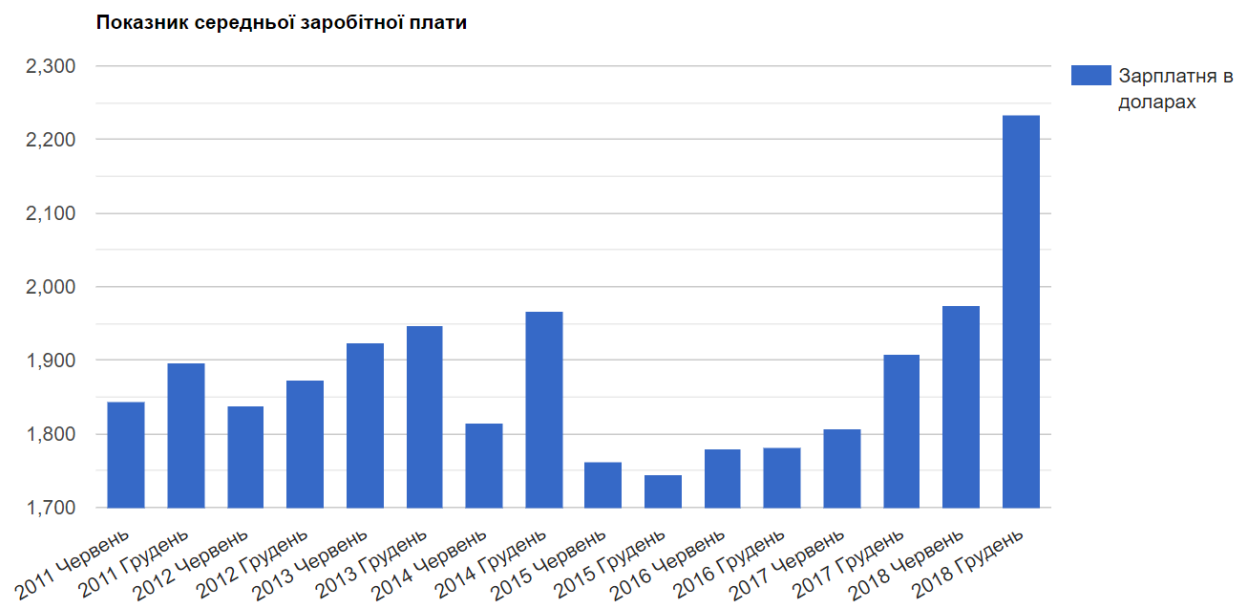


Рисунок 5.3 – Відображення графіку зміни середньої зарплатні

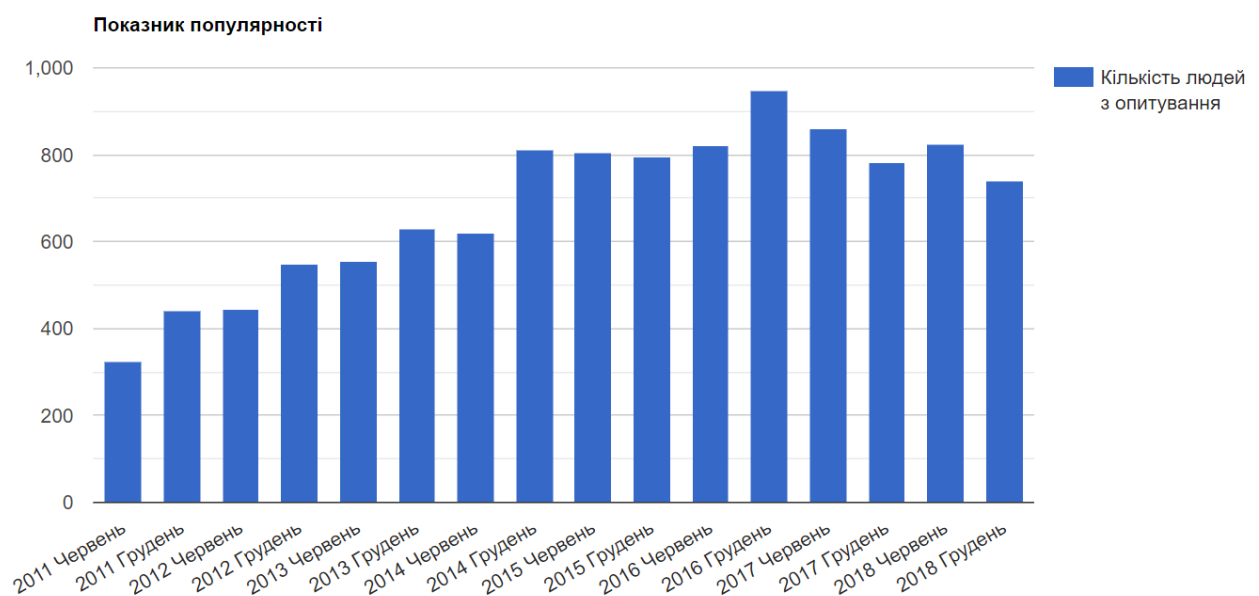


Рисунок 5.4 – Відображення графіку зміни популярності

Для виконання прогнозування, користувач відкриває програму з налаштованою нейронною мережею в MATLAB, і отримує результати прогнозу. На даний момент мережа натренована прогнозувати лише зарплатню для всієї ІТ-індустрії. Результат роботи мережі зображено на рисунку 5.5.

c = 1.0e+03 *

Columns 1 through 6

1.6713 1.8510 1.8270 1.7954 1.7884 1.8589

Columns 7 through 12

1.7789 1.7872 1.7087 1.6460 1.6617 1.6654

Columns 13 through 16

1.6773 1.7372 1.8400 1.9909

c2 = 2.0920e+03

Рисунок 5.5 – Результат роботи нейронної мережі

5.2 Випробування програмного продукту

5.2.1 Мета випробування

Мета випробування – встановити чи відповідає програма поставленим у технічному завданні вимогам.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

					ДП ІС-5119.1181-с.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Процедура проведення випробувань та їх результати занесені у таблицю 5.1.

Таблиця 5.1 Зміст та результати випробувань.

Дія	Очікуваний результат	Результат тесту
Ввести html адресу застосунку в строку пошуку браузеру та натиснути пошук	Відкрита сторінка “Введення запиту” на ній зображено шість спадних списків з наступними підписами: <ul style="list-style-type: none"> – “Місто” обрана опція: “вся Україна”; – “Посада” обрана опція: “будь-яка”; – “Мова програмування” обрана опція: “будь-яка”; – “Рівень Англійського” обрана опція: “будь-який”; – “досвід роботи” – Кнопка “Відправити запит”. 	пройдений

Продовження таблиці 5.1

Обрати зі списку професій ті які стосуються QA	Відкрита сторінка “Введення запити” – Спадний список підписаний “Мова програмування” заміщується списком підписаним “Спеціалізація” Обрана опція: будь-яка	пройдений
Обрати зі списку професій ті які стосуються написання коду	Відкрита сторінка “Введення запити” – Спадний список підписаний “Спеціалізація” заміщується списком підписаним “Мова програмування” Обрана опція: будь-яка	пройдений
Натиснути на кнопку “Відправити запит”	Відкрита сторінка “Введення запити” – Очікування відповіді від серверу, допустимий інтервал часу – чотири секунди – Під кнопкою “Відправити запит” з’являються два графіки які відповідають запити	пройдений

ЗАГАЛЬНІ ВИСНОВКИ

В результаті виконання дипломної роботи були отримані наступні результати:

- спроектовано БД що зберігає дані опитувань взяті з сайту github.com [2], забезпечивши можливість подальшої обробки великих об'ємів інформації;
- реалізовано систему аналізу, здатну до обробки великої кількості результатів опитувань і створення на їх основі статистичної інформації що описує тренди зміни ЗП, що в подальшому використовувалась для відображення користувачеві та в якості вхідних даних для навчання нейронної мережі;
- використано інструмент Google Charts для відображення даних, що дозволило досягнути мети проекту в візуалізації даних;
- спроектовано та навчено нейронну мережу на основі даних системи аналізу, що довело можливість використання нейронної мережі для прогнозування ЗП;
- введення додаткової ознаки в навчальну вибірку дозволило збільшити точність прогнозування.

ПЕРЕЛІК ПОСИЛАНЬ

1. ІТ-ІНДУСТРІЯ В УКРАЇНІ ПОСІЛА ДРУГЕ МІСЦЕ В ЕКСПОРТІ ПОСЛУГ І ЗАЛИШАЄТЬСЯ НАЙБІЛЬШ ОПЛАЧУВАНОЮ ГАЛУЗЗЮ 14.01.2019 [Електронний ресурс]: Україна молода – Режим доступу: <https://www.umoloda.kiev.ua/number/3407/188/129512/>
- Дата доступу: 14.01.2019.
2. Исходные данные опроса в CSV [Електронний ресурс]. – Режим доступу: <https://github.com/devua/csv/tree/master/salaries> - Дата доступу: 20.01.2019.
3. Зарплаты разработчиков [Електронний ресурс]. – Режим доступу: <https://jobs.dou.ua/salaries/> – Дата доступу: 21.01.2019
4. Альтернативный виджет [Електронний ресурс]. – Режим доступу: <http://doustatistic.byethost7.com/> - Дата доступу: 20.01.2019.
5. Денормализация данных [Електронний ресурс]. - Режим доступу: <https://ruhighload.com/Денормализация+данных> – Дата доступу: 30.05.2018
6. Схемы звезда и снежинка [Публікація]. – Режим доступу: <http://new-techs.ru/sxemy-zvezda-i-snezhinka.html> – Дата доступу: 31.01.2016
7. Галушкин А. И. Теория нейронных сетей. Кн.1: Учебное пособие для вузов. / А. И. Галушкин. // Издательское предприятие редакции журнала «Радиотехника». – 2000. – С. 215
8. Мкртчян С. О. Нейроны и нейронные сети / С. О. Мкртчян. – М.: Энергия, 1970. – 230 с
9. Хайкин С. Нейронные сети / С. Хайкин. – М.: Вильямс, 2006.–1103с.
10. ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ ДЛЯ ПРОГНОЗУВАННЯ У ФІНАНСОВІЙ СФЕРІ [Електронний ресурс]. – Режим доступу: <http://ape.fmm.kpi.ua/article/download/102584/97660>
- Дата доступу: 01.01.2017

11. Применение нейросетей в распознавании изображений [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/74326/> – Дата доступа: 05.11.2009
12. Герасименко Н. А. Нейросетевые технологии в анализе фондового рынка [Электронный ресурс] / 7. Герасименко Н. А.. – 1998. – Режим доступа: http://fakit.ru/main_dsp.php?top_id=1086
13. ОБЛАСТЬ ПРИМЕНЕНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ И ПЕРСПЕКТИВЫ ИХ РАЗВИТИЯ / А. И. Галушкин. // Научный журнал КубГАУ, №27(3), март 2007
14. Новиков В.А. Организация и обучение искусственных нейронных сетей: Экспериментальное учеб. пособие. / В.А.Новиков, Л.В.Калацкая, В.С.Садов // Минск: БГУ, 2003. – 72 с.
15. MVC — модель-представление-контроллер [Электронный ресурс]. - Режим доступа: <https://web-creator.ru/articles/mvc> - Дата доступа: 14.02.2017
16. Как работает Java машина (JVM) [Электронный ресурс]. - Режим доступа: <https://idurdyev.com/kak-rabotaet-jvm> - Дата доступа: 21.11.2016
17. Interactive charts for browsers and mobile devices. [Электронный ресурс]. - Режим доступа: <https://developers.google.com/chart/?hl=uk> - Дата доступа: 30.05.2018
18. Знакомство с Thymeleaf [Электронный ресурс]. - Режим доступа: <https://habr.com/ru/post/350864/> - Дата доступа: 11.03.2018
19. JPA работа с базой данных [Электронный ресурс]. - Режим доступа: <https://devcolibri.com/jpa-работа-с-базой-данных-часть-1/> - Дата доступа: 18.05.2013

ДОДАТОК А

*Тексти програмного коду***Net.m**

```
%Prognoz serednoi zarobitnoi platy
close all
clear all
clc;
x1=0:15;
y1 = [1695  1863 1857 1839 1806 1878 1813 1855 1717 1721 1670 1658 1669
1750 1840 2000];
plot(x1,y1);
title('Pokaznyk serednoi zarobitnoi platy');
xlabel('Roky 0- 2011; 15 - 2018');
ylabel('Riven zarobitnoi platy');
% Navchalna vybirka
x = [1   2   3   4   5   6   7   8   9  10 11 12 13 14 15;
      2   2   1   1   1   2   1   2   1   2   1   1   2   2   2];
y = [1695  1863 1857 1839 1806 1878 1813 1855 1717 1721 1670 1658 1669
1750 1840];

net=newff(x,y,[12,1],{'logsig', 'purelin'}, 'trainlm');
net.trainparam.show=25;
net.trainparam.lr= 0.1;
net.trainparam.epochs=500;
net.trainparam.goal=0.001;
net=train(net, x, y);
c = sim(net,x);
c2 = sim(net,[16;2]);
display(c);
display(c2);
```

SearchController.java

```
@Controller
public class SearchController {
```

```
@Autowired
private FormDataService formDataService;

@Autowired
private GraphBuilderService graphBuilderService;

@Autowired
private GoogleAPIService googleAPIService;

@GetMapping
public String showIndex(Model model) {
    model.addAttribute("formData", formDataService.getFormData());
    return "index";
}

@PostMapping
public String showResponse(@RequestParam(required = false) String city,
                           @RequestParam(required = false) String position,
                           @RequestParam(required = false) String language,
                           @RequestParam(required = false) String specialization,
                           @RequestParam(required = false) String englishLevel,
                           @RequestParam(required = false) String educationLevel,
                           @RequestParam(required = false) int minimalExperience,
                           @RequestParam(required = false) int maximalExperience,
                           Model model) {

    SearchParams searchParams = new SearchParams();
    searchParams.setCity(city);
    searchParams.setPosition(position);
    searchParams.setLanguage(language);
    searchParams.setSpecialization(specialization);
    searchParams.setEnglishLevel(englishLevel);
    searchParams.setEducationLevel(educationLevel);
    searchParams.setMinimalExperience(minimalExperience);
    searchParams.setMaximalExperience(maximalExperience);

    List<Point> points = graphBuilderService.getPoints(searchParams);
    List<GoogleAPIPoint> apiPoints = googleAPIService.toAPIPoints(points);
    if (apiPoints == null) model.addAttribute("errorMessage", "Недостатньо даних
для аналізу");

    model.addAttribute("points", apiPoints);
}
```

```
        model.addAttribute("formData", formDataService.getFormData());

        return "index";
    }
}
```

FormData.java

```
@Entity
public class FormData {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;

    @ElementCollection(fetch = FetchType.EAGER)
    @CollectionTable(name = "cities", joinColumns = @JoinColumn(name = "formData_id"))
    @Column(name = "city")
    private Set<String> cities;

    @ElementCollection(fetch = FetchType.EAGER)
    @CollectionTable(name = "educationLevels", joinColumns = @JoinColumn(name = "formData_id"))
    @Column(name = "educationLevel")
    private Set<String> educationLevels;

    @ElementCollection(fetch = FetchType.EAGER)
    @CollectionTable(name = "englishLevels", joinColumns = @JoinColumn(name = "formData_id"))
    @Column(name = "englishLevel")
    private Set<String> englishLevels;

    @ElementCollection(fetch = FetchType.EAGER)
    @CollectionTable(name = "genders", joinColumns = @JoinColumn(name = "formData_id"))
    @Column(name = "gender")
    private Set<String> genders;

    @ElementCollection(fetch = FetchType.EAGER)
    @CollectionTable(name = "positions", joinColumns = @JoinColumn(name = "formData_id"))
    @Column(name = "position")
    private Set<String> positions;
```

```
@ElementCollection(fetch = FetchType.EAGER)
@CollectionTable(name = "programmingLanguages", joinColumns = @JoinColumn(name =
"formData_id"))
@Column(name = "programmingLanguage")
private Set<String> programmingLanguages;

@ElementCollection(fetch = FetchType.EAGER)
@CollectionTable(name = "specializations", joinColumns = @JoinColumn(name =
"formData_id"))
@Column(name = "specialization")
private Set<String> specializations;

@ElementCollection(fetch = FetchType.EAGER)
@CollectionTable(name = "subjectAreas", joinColumns = @JoinColumn(name =
"formData_id"))
@Column(name = "subjectArea")
private Set<String> subjectAreas;

@ElementCollection(fetch = FetchType.EAGER)
@CollectionTable(name = "surveyDates", joinColumns = @JoinColumn(name =
"formData_id"))
@Column(name = "surveyDate")
private Set<String> surveyDates;

@Override
public String toString() {
    return "FormData{" +
        "\ncities=" + cities +
        ", \neducationLevels=" + educationLevels +
        ", \nenglishLevels=" + englishLevels +
        ", \npositions=" + positions +
        ", \nprogrammingLanguages=" + programmingLanguages +
        ", \nspecializations=" + specializations +
        ", \nsurveyDates=" + surveyDates +
        '}';
}

public Set<String> getCities() {
    return new TreeSet<>(cities);
}

public void setCities(Set<String> cities) {
    this.cities = cities;
}
```



```
    }

    public Set<String> getEducationLevels() {
        return new TreeSet<>(educationLevels);
    }

    public void setEducationLevels(Set<String> educationLevels) {
        this.educationLevels = educationLevels;
    }

    public Set<String> getEnglishLevels() {
        return new TreeSet<>(englishLevels);
    }

    public void setEnglishLevels(Set<String> englishLevels) {
        this.englishLevels = englishLevels;
    }

    public Set<String> getGenders() {
        return new TreeSet<>(genders);
    }

    public void setGenders(Set<String> genders) {
        this.genders = genders;
    }

    public Set<String> getPositions() {
        return new TreeSet<>(positions);
    }

    public void setPositions(Set<String> positions) {
        this.positions = positions;
    }

    public Set<String> getProgrammingLanguages() {
        return new TreeSet<>(programmingLanguages);
    }

    public void setProgrammingLanguages(Set<String> programmingLanguages) {
        this.programmingLanguages = programmingLanguages;
    }

    public Set<String> getSpecializations() {
        return new TreeSet<>(specializations);
    }
```

```
    }

    public void setSpecializations(Set<String> specializations) {
        this.specializations = specializations;
    }

    public Set<String> getSubjectAreas() {
        return new TreeSet<>(subjectAreas);
    }

    public void setSubjectAreas(Set<String> subjectAreas) {
        this.subjectAreas = subjectAreas;
    }

    public Set<String> getSurveyDates() {
        return new TreeSet<>(surveyDates);
    }

    public void setSurveyDates(Set<String> surveyDates) {
        this.surveyDates = surveyDates;
    }
}
```

GoogleAPIPoint.java

```
public class GoogleAPIPoint {
    private String date;
    private int averageSalary;
    private int surveyQuantity;

    public GoogleAPIPoint(Point point){
        date = point.getDate();
        averageSalary = (int)point.getAverageSalary();
        surveyQuantity = (int)point.getSurveysInDate();
    }

    public String getDate() {
        return date;
    }

    public int getAverageSalary() {
        return averageSalary;
    }
}
```

```
    }

    public int getSurveyQuantity() {
        return surveyQuantity;
    }
}
```

Point.java

```
public class Point {
    private String date;
    private double averageSalary;
    private double totalSalary;
    private double surveysInDate;

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }

    public double getAverageSalary() {
        return averageSalary;
    }

    public void setAverageSalary(double averageSalary) {
        this.averageSalary = averageSalary;
    }

    public double getTotalSalary() {
        return totalSalary;
    }

    public void setTotalSalary(double totalSalary) {
        this.totalSalary = totalSalary;
    }

    public double getSurveysInDate() {
        return surveysInDate;
    }
}
```

```
    }

    public void setSurveysInDate(double surveysInDate) {
        this.surveysInDate = surveysInDate;
    }

    @Override
    public String toString() {
        return "\nPoint{" +
            "date='" + date + '\'' +
            ", averageSalary=" + averageSalary +
            ", totalSalary=" + totalSalary +
            ", surveysInDate=" + surveysInDate +
            '\'';
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Point point = (Point) o;
        return Objects.equals(date, point.date);
    }

    @Override
    public int hashCode() {
        return Objects.hash(date);
    }
}
```

SearchParams.java

```
public class SearchParams {
    private String city;
    private String position;
    private String englishLevel;
    private String language;
    private String specialization;
    private String educationLevel;
    private int minimalExperience;
    private int maximalExperience;
```

```
public String getCity() {
    return city;
}

public void setCity(String city) {
    this.city = city;
}

public String getPosition() {
    return position;
}

public void setPosition(String position) {
    this.position = position;
}

public String getEnglishLevel() {
    return englishLevel;
}

public void setEnglishLevel(String englishLevel) {
    this.englishLevel = englishLevel;
}

public String getLanguage() {
    return language;
}

public void setLanguage(String language) {
    this.language = language;
}

public String getSpecialization() {
    return specialization;
}

public void setSpecialization(String specialization) {
    this.specialization = specialization;
}

public String getEducationLevel() {
    return educationLevel;
}
```

```
public void setEducationLevel(String educationLevel) {
    this.educationLevel = educationLevel;
}

public int getMinimalExperience() {
    return minimalExperience;
}

public void setMinimalExperience(int minimalExperience) {
    this.minimalExperience = minimalExperience;
}

public int getMaximalExperience() {
    return maximalExperience;
}

public void setMaximalExperience(int maximalExperience) {
    this.maximalExperience = maximalExperience;
}

@Override
public String toString() {
    return "SearchParams{" +
        "city='" + city + '\'' +
        ", position='" + position + '\'' +
        ", englishLevel='" + englishLevel + '\'' +
        ", language='" + language + '\'' +
        ", specialization='" + specialization + '\'' +
        ", educationLevel='" + educationLevel + '\'' +
        ", minimalExperience=" + minimalExperience +
        ", maximalExperience=" + maximalExperience +
        '}';
}
}
```

Survey.java

```
@Entity
public class Survey {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private long id;
```

```
private String position;

private String programmingLanguage;

private String specialization;

private float experience;

private float salary;

private String city;

private String gender;

private String education;

private String stillStudent;

private String englishLevel;

private String subjectArea;

private Date surveyDate;

public void setSalary(float salary) {
    this.salary = salary;
}

public String getStillStudent() {
    return stillStudent;
}

public Date getSurveyDate() {
    return surveyDate;
}

public void setSurveyDate(Date surveyDate) {
    this.surveyDate = surveyDate;
}

public long getId() {
    return id;
}
```

```
    }

    public void setId(long id) {
        this.id = id;
    }

    public String getPosition() {
        return position;
    }

    public void setPosition(String position) {
        this.position = position;
    }

    public String getProgrammingLanguage() {
        return programmingLanguage;
    }

    public void setProgrammingLanguage(String programmingLanguage) {
        this.programmingLanguage = programmingLanguage;
    }

    public String getSpecialization() {
        return specialization;
    }

    public void setSpecialization(String specialization) {
        this.specialization = specialization;
    }

    public double getExperience() {
        return experience;
    }

    public void setExperience(float experience) {
        this.experience = experience;
    }

    public float getSalary() {
        return salary;
    }

    public void setSalary(int salary) {
        this.salary = salary;
    }
```



```
}

public String getCity() {
    return city;
}

public void setCity(String city) {
    this.city = city;
}

public String getGender() {
    return gender;
}

public void setGender(String gender) {
    this.gender = gender;
}

public String getEducation() {
    return education;
}

public void setEducation(String education) {
    this.education = education;
}

public String isStillStudent() {
    return stillStudent;
}

public void setStillStudent(String stillStudent) {
    this.stillStudent = stillStudent;
}

public String getEnglishLevel() {
    return englishLevel;
}

public void setEnglishLevel(String englishLevel) {
    this.englishLevel = englishLevel;
}

public String getSubjectArea() {
    return subjectArea;
}
```

```
    }

    public void setSubjectArea(String subjectArea) {
        this.subjectArea = subjectArea;
    }

    @Override
    public String toString() {
        return "Survey{" +
            "id=" + id +
            ",\tposition='" + position + '\'' +
            ",\tprogrammingLanguage='" + programmingLanguage + '\'' +
            ",\tspecialization='" + specialization + '\'' +
            ",\texperience=" + experience +
            ",\tsalary=" + salary +
            ",\tcity='" + city + '\'' +
            ",\tgender='" + gender + '\'' +
            ",\teducation='" + education + '\'' +
            ",\tstillStudent='" + stillStudent + '\'' +
            ",\tenglishLevel='" + englishLevel + '\'' +
            ",\tsubjectArea='" + subjectArea + '\'' +
            '}';
    }
}
```

FormDataRepository.java

```
public interface FormDataRepository extends CrudRepository<FormData, Long>,
JpaSpecificationExecutor<FormData> {
}
```

SurveyRepository.java

```
public interface SurveyRepository extends CrudRepository<Survey, Long>,
JpaSpecificationExecutor<Survey> {
}
```

DateService.java

```
@Service
public class DateService {
    public String dateToString(Date date){
        return new SimpleDateFormat("yyyy-MM-dd").format(date);
    }

    public String transformDate(String date){
        String[] parts = date.split("[^\\d]");
        if (parts[1].equals("06")){
            parts[1] = " Червень";
        } else {
            parts[1] = " Грудень";
        }

        return parts[0]+parts[1];
    }

    public List<Point> transformDates(List<Point> points){
        for (Point p : points){
            p.setDate(transformDate(p.getDate()));
        }
        return points;
    }
}
```

FormDataService.java

```
@Service
public class FormDataService {

    @Autowired
    private FormDataRepository formDataRepository;

    @Autowired
    private SurveyService surveyService;

    @Autowired DateService dateService;

    public FormData getFormData(){
        if (formDataRepository.count() > 0){
            return getFormDataFromRepo();
        } else {
```

```
        refreshFormData();
        return getFormDataFromRepo();
    }
}

private FormData getFormDataFromRepo(){
    return formDataRepository.findAll().iterator().next();
}

private void clearRepo(){
    formDataRepository.deleteAll();
}

public void refreshFormData() {
    clearRepo();
    FormData formData = new FormData();

    Set<String> cities = new TreeSet<>();
    Set<String> educationLevels = new TreeSet<>();
    Set<String> englishLevels = new TreeSet<>();
    Set<String> genders = new TreeSet<>();
    Set<String> positions = new TreeSet<>();
    Set<String> programmingLanguages = new TreeSet<>();
    Set<String> specializations = new TreeSet<>();
    Set<String> subjectAreas = new TreeSet<>();
    Set<String> surveyTimes = new TreeSet<>();

    for (Survey survey : surveyService.getAllSurveys()) {
        cities.add(survey.getCity());
        educationLevels.add(survey.getEducation());
        englishLevels.add(survey.getEnglishLevel());
        genders.add(survey.getGender());
        for(String pos: survey.getPosition().split(" / ")){
            positions.add(pos);
        }
        programmingLanguages.add(survey.getProgrammingLanguage());
        programmingLanguages.remove("");
        specializations.add(survey.getSpecialization());
        specializations.remove("");
        for (String sa : survey.getSubjectArea().split(",")){
            subjectAreas.add(sa);
        }
        subjectAreas.remove("");
        subjectAreas.remove("na");
    }
}
```

```
        surveyTimes.add(dateService.dateToString(survey.getSurveyDate()));
    }

    formData.setCities(cities);
    formData.setEducationLevels(educationLevels);
    formData.setEnglishLevels(englishLevels);
    formData.setGenders(genders);
    formData.setPositions(positions);
    formData.setProgrammingLanguages(programmingLanguages);
    formData.setSpecializations(specializations);
    formData.setSubjectAreas(subjectAreas);
    formData.setSurveyDates(surveyTimes);

    formDataRepository.save(formData);
}
}
```

GoogleAPIService.java

```
@Service
public class GoogleAPIService {
    public List<GoogleAPIPoint> toAPIPoints(List<Point> points){
        if (points == null) return null;
        List<GoogleAPIPoint> apiPoints = new ArrayList<>();
        for (Point p : points){
            apiPoints.add(new GoogleAPIPoint(p));
        }
        return apiPoints;
    }
}
```

GraphBuilderService.java

```
@Service
public class GraphBuilderService {

    @Autowired
    private SurveyService surveyService;

    @Autowired
    private MathService mathService;
```

```
@Autowired
private DateService dateService;

public List<Point> getPoints(SearchParams searchParams) {
    List<Survey> surveys = surveyService.getSurveysWithParams(searchParams);
    Map<String, Point> pointsMap = new HashMap<>();
    for (Survey survey : surveys) {
        String currentDate = dateService.dateToString(survey.getSurveyDate());
        if (pointsMap.containsKey(currentDate)) {
            Point currentPoint = pointsMap.get(currentDate);
            currentPoint.setTotalSalary(currentPoint.getTotalSalary()
survey.getSalary());
            currentPoint.setSurveysInDate(currentPoint.getSurveysInDate() + 1);
            pointsMap.put(currentDate, currentPoint);
        } else {
            Point newPoint = new Point();
            newPoint.setDate(currentDate);
            newPoint.setTotalSalary(survey.getSalary());
            pointsMap.put(currentDate, newPoint);
        }
    }
    List<Point> result = new ArrayList<>();
    for (Point point : pointsMap.values()) {
        point.setAverageSalary(mathService.getAverage(point.getTotalSalary(),
point.getSurveysInDate()));
        result.add(point);
    }
    result.sort(Comparator.comparing(Point::getDate));
    if (result.size() == 0) return null;

    dateService.transformDates(result);

    return result;
}
```

MathService.java

```
@Service
public class MathService {
```

```
public double getAverage(double sum, double quantity){
    if ((int)quantity == 0) return 0;
    else return sum/quantity;
}
}
```

SurveyService.java

```
@Service
public class SurveyService {
    @Autowired
    private SurveyRepository surveyRepository;

    @Autowired
    private SurveySpecifications surveySpecifications;

    public Iterable<Survey> getAllSurveys() {
        return surveyRepository.findAll();
    }

    public List<Survey> getSurveysWithParams(SearchParams searchParams){
        List<Survey> result = surveyRepository.findAll(surveySpecifications.
            withCity(searchParams.getCity()).and(surveySpecifications.
            withEducation(searchParams.getEducationLevel()).and(surveySpecifications.
            withEnglishLevel(searchParams.getEnglishLevel()).and(surveySpecifications.
            withExperienceLessThan(searchParams.getMaximalExperience()).and(surveySpecifications.
            withExperienceMoreThan(searchParams.getMinimalExperience()).and(surveySpecifications.
            withPosition(searchParams.getPosition()).and(surveySpecifications.
            withProgrammingLanguage(searchParams.getLanguage()).and(surveySpecifications.
            withSpecialization(searchParams.getSpecialization()))))))));
        result.sort(Comparator.comparing(Survey::getSurveyDate));
        return result;
    }
}
```

SurveySpecifications.java

```
@Service
public class SurveySpecifications {
```

```
public Specification<Survey> withCity(String var) {
    return (root, cq, cb) -> cb.like(cb.lower(root.get("city")), "%" + var.toLowerCase() +
"%");
}

public Specification<Survey> withEducation(String var) {
    return (root, cq, cb) -> cb.like(cb.lower(root.get("education")), "%" +
var.toLowerCase() + "%");
}

public Specification<Survey> withEnglishLevel(String var) {
    return (root, cq, cb) -> cb.like(cb.lower(root.get("englishLevel")), "%" +
var.toLowerCase() + "%");
}

public Specification<Survey> withExperienceLessThan(int val) {
    return (root, cq, cb) -> cb.lessThanOrEqualTo(root.get("experience"), val);
}

public Specification<Survey> withExperienceMoreThan(int val) {
    return (root, cq, cb) -> cb.greaterThanOrEqualTo(root.get("experience"), val);
}

public Specification<Survey> withPosition(String var) {
    return (root, cq, cb) -> cb.like(root.get("position"), "%" + var + "%");
}

public Specification<Survey> withProgrammingLanguage(String var) {
    if (var == ""){
        return (root, cq, cb) -> cb.like(root.get("programmingLanguage"), "%" + var +
"%");
    } else {
        return (root, cq, cb) -> cb.like(cb.lower(root.get("programmingLanguage")),
var.toLowerCase());
    }
}

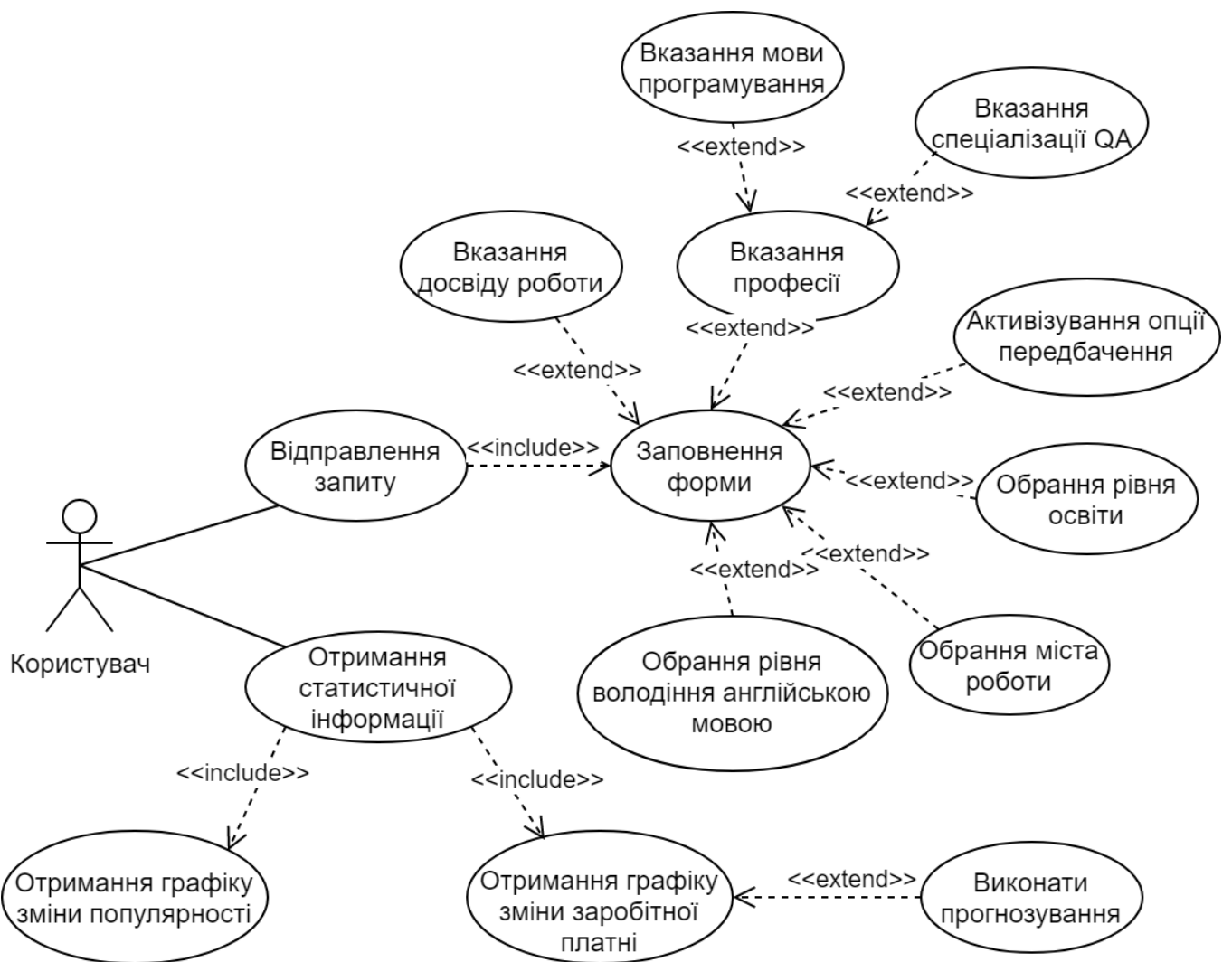
public Specification<Survey> withSpecialization(String var) {
    return (root, cq, cb) -> cb.like(cb.lower(root.get("specialization")), "%" +
var.toLowerCase() + "%");
}
}
```


ItMarketAnalyticsApplication.java

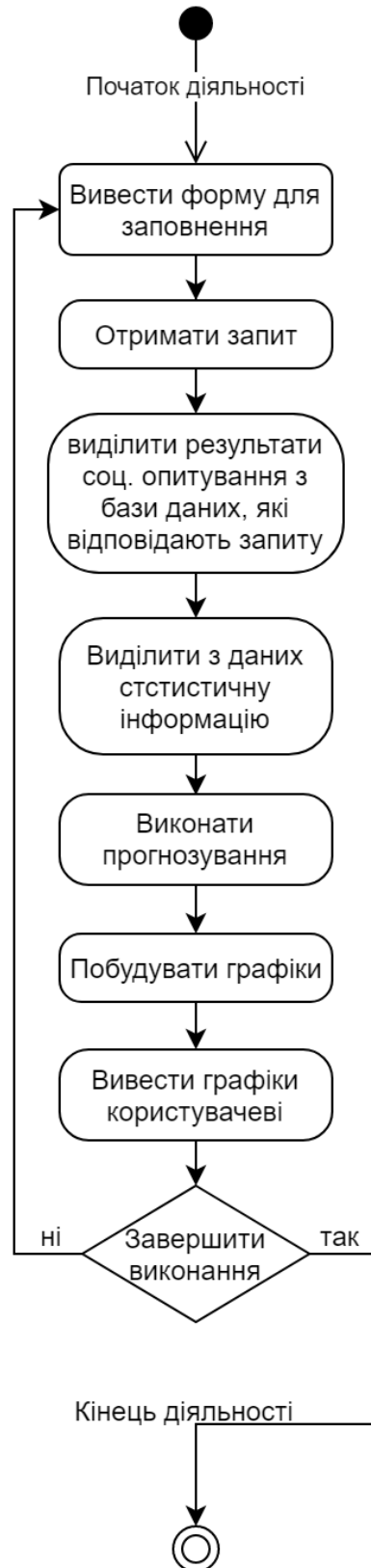
```
@SpringBootApplication
public class ItMarketAnalyticsApplication {

    public static void main(String[] args) {
        SpringApplication.run(ItMarketAnalyticsApplication.class, args);
    }

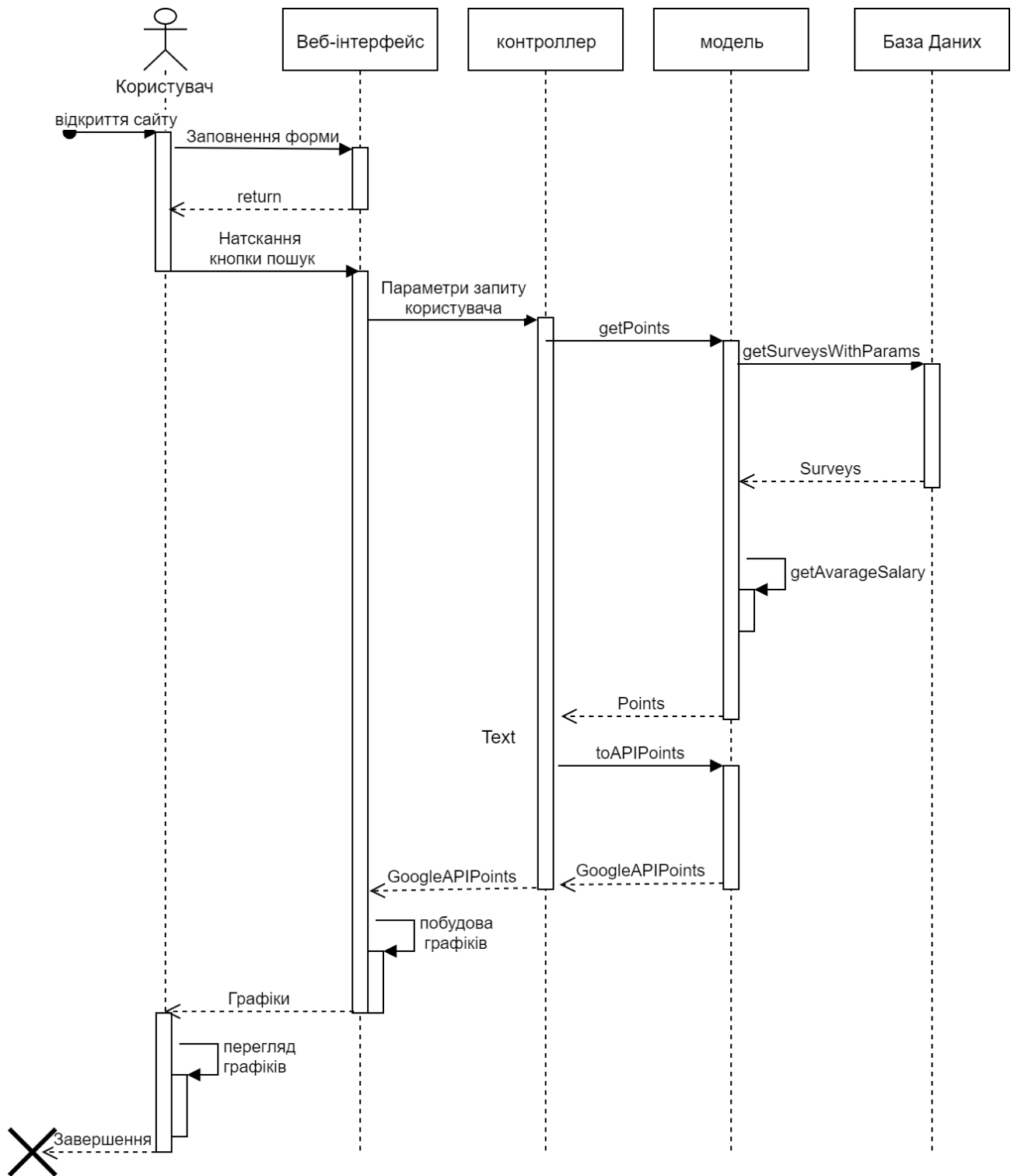
}
```



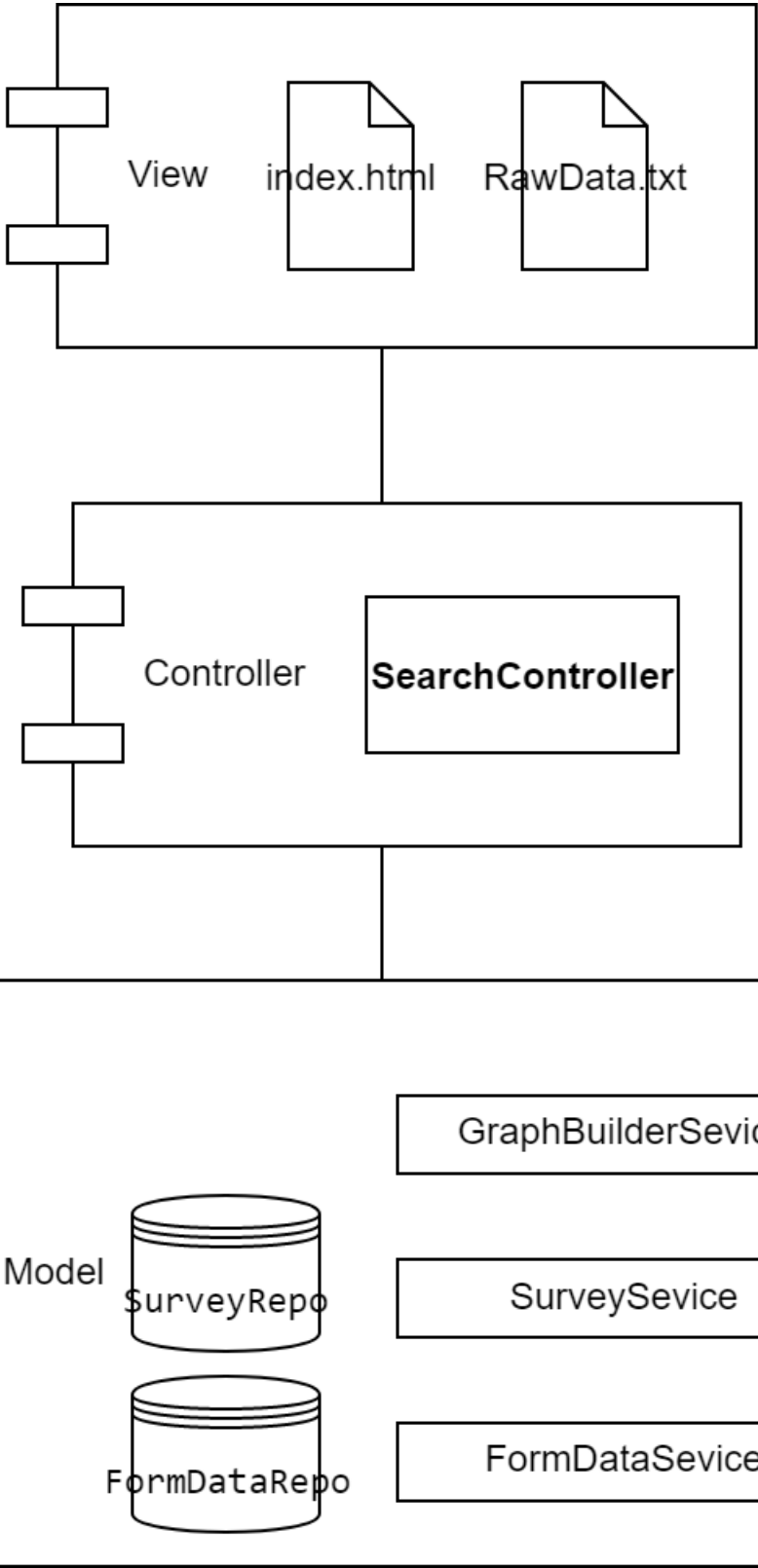
					ДП ІС-5119.1181-с.ССВ									
					Схема структурна варіантів використань									
Зм.	Арк.	№ документа	Підпис	Дата	Комплекс задач аналізу ринку праці фахівців з інформаційних технологій									
Розробив		Пономаренко А.О.												
Перевірив		Звенігородський О.С.												
Т. кон.					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51									
Н. кон.		Москаленко Н.В.												
Затвердив		Звенігородський О.С.												
					Літера									
					Маса									
					Масштаб									
					Аркуш 1									
					Аркушів 1									



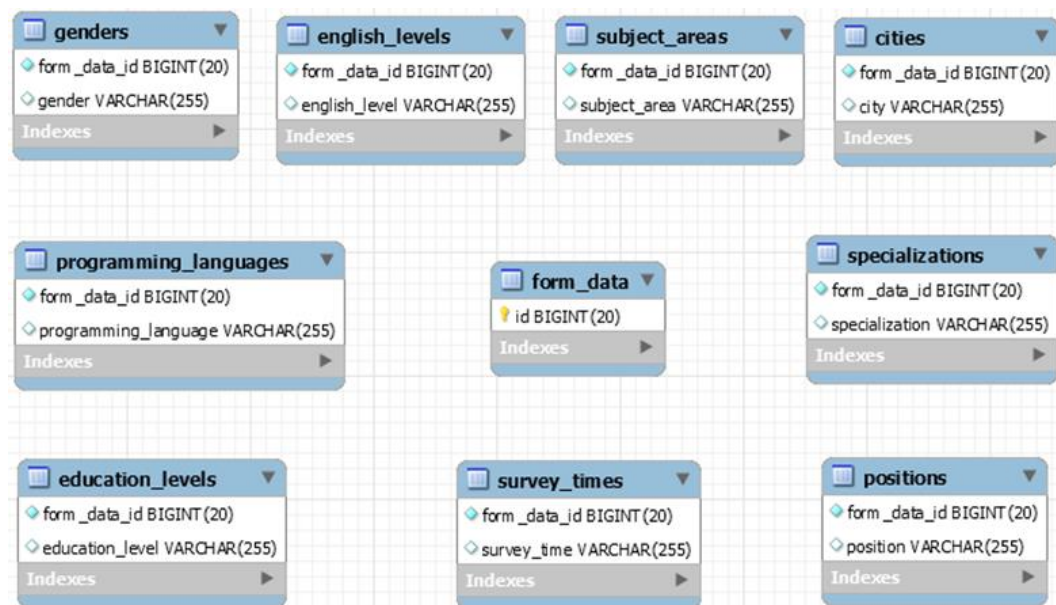
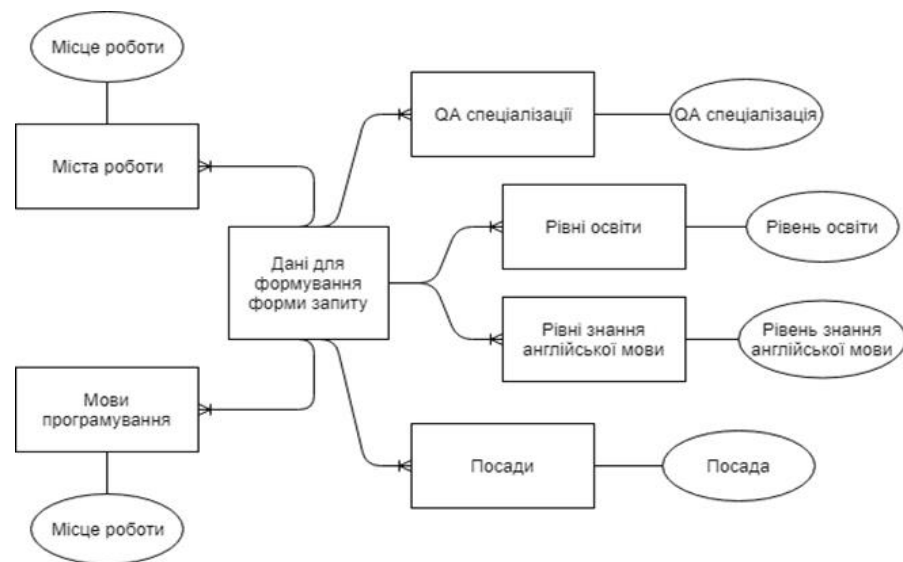
					ДП ІС-5119.1181-с.ССД							
					Схема структурна діяльності	Літера			Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив	Пономаренко А.О.											
Перевірив	Звенігородський О.С.					Аркуш 1			Аркушів 1			
Т. кон.					Комплекс задач аналізу ринку праці фахівців з інформаційних технологій	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51						
Н. кон.	Москаленко Н.В											
Затвердив	Звенігородський О.С.											



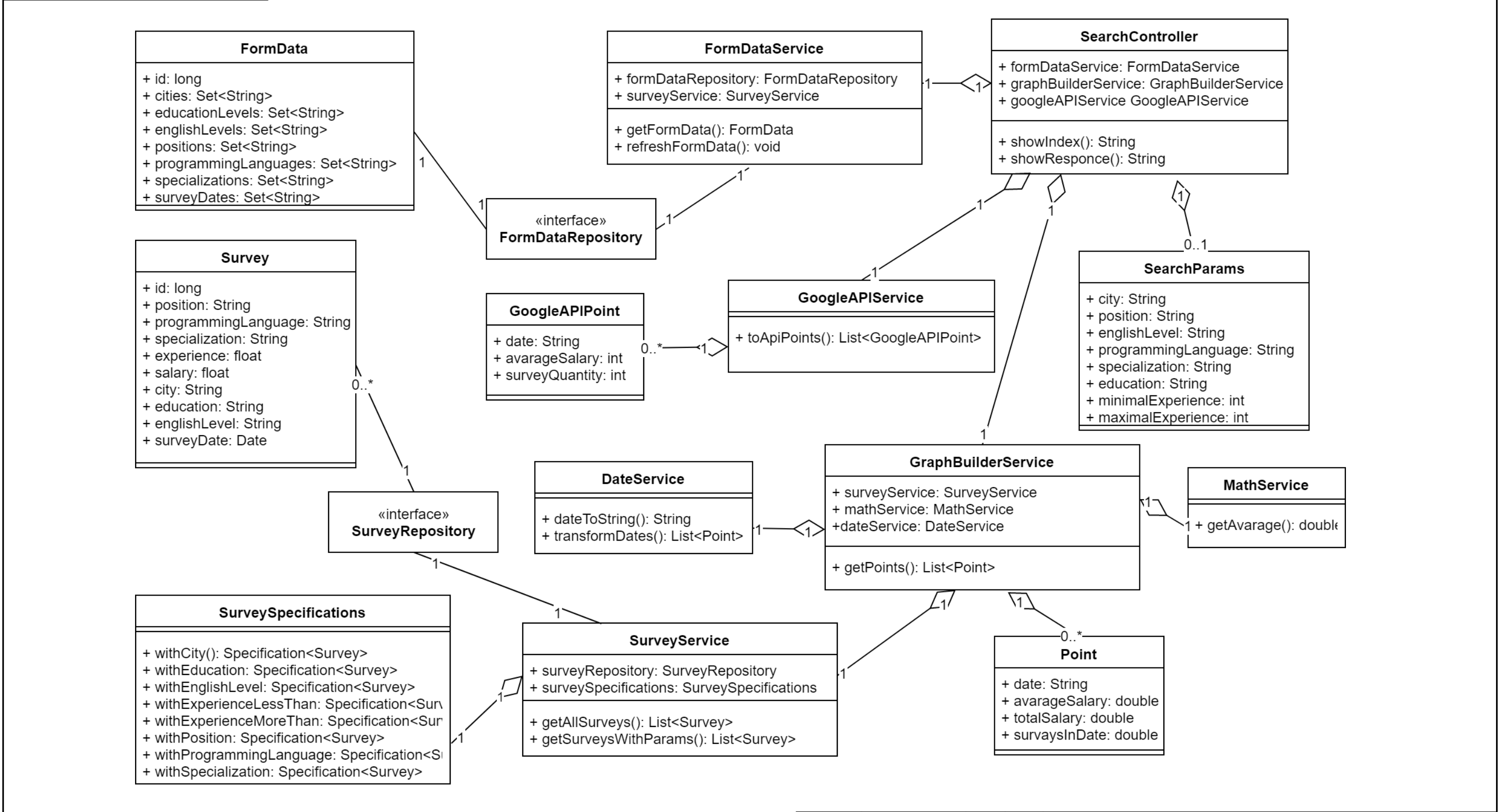
					ДП ІС-5119.1181-с.ССП				
					Схема структурна послідовності				
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Пономаренко А.О.							
Перевірив		Звенігородський О.С.							
Т. кон.									
Н. кон.		Москаленко Н.В.							
Затвердив		Звенігородський О.С.							
					Комплекс задач аналізу ринку праці фахівців з інформаційних технологій				
					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51				



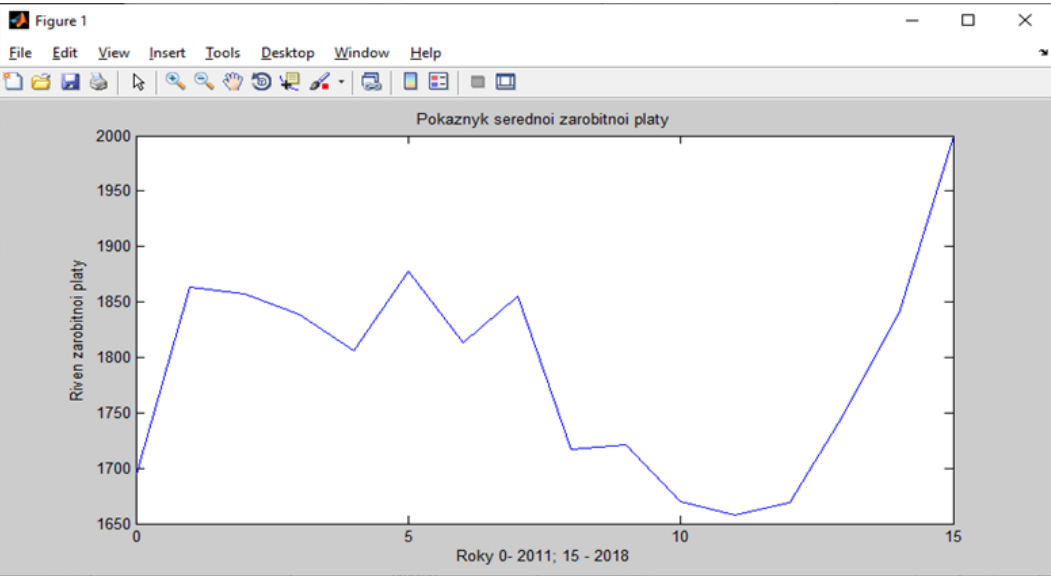
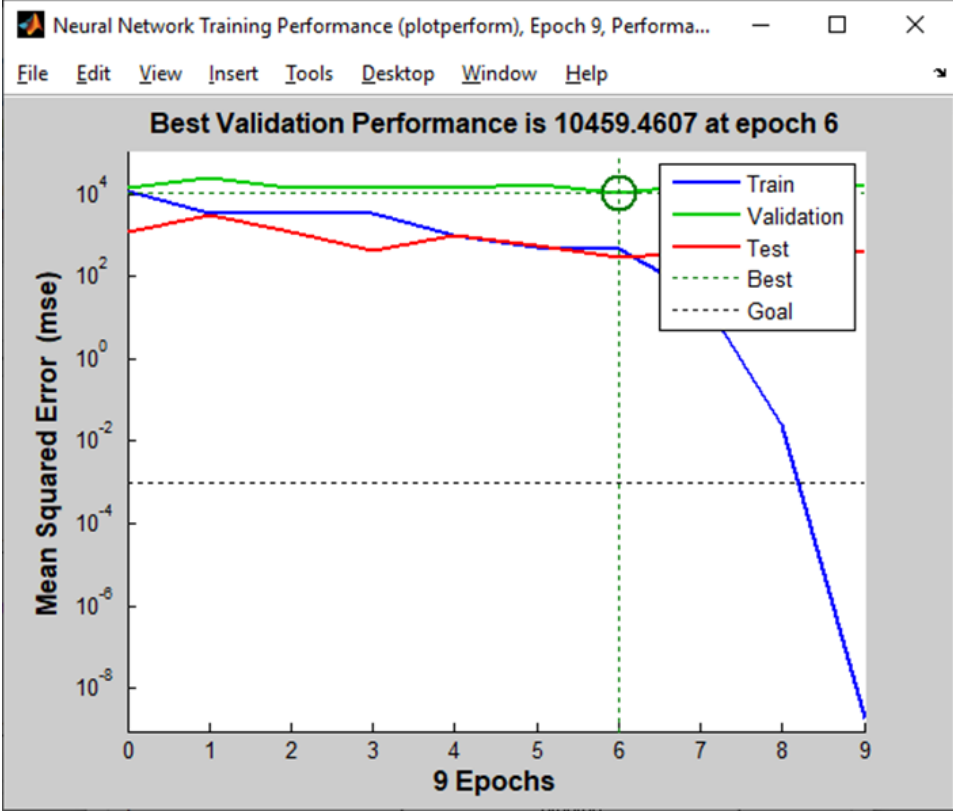
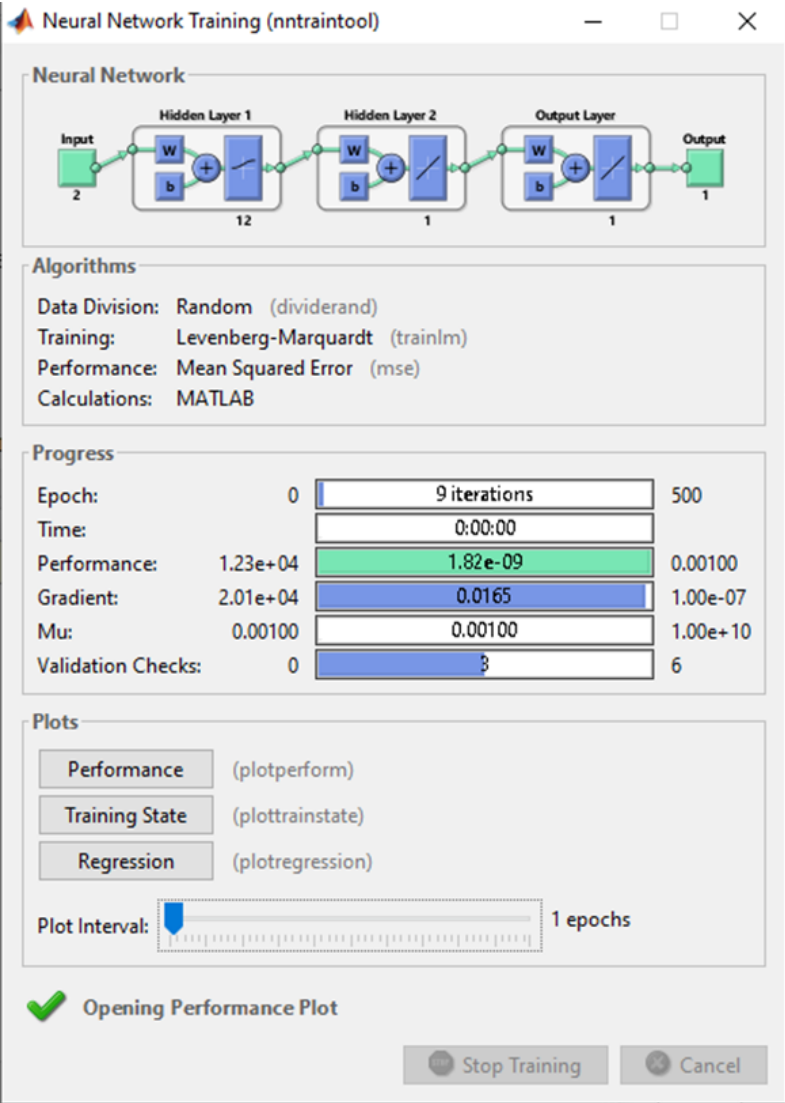
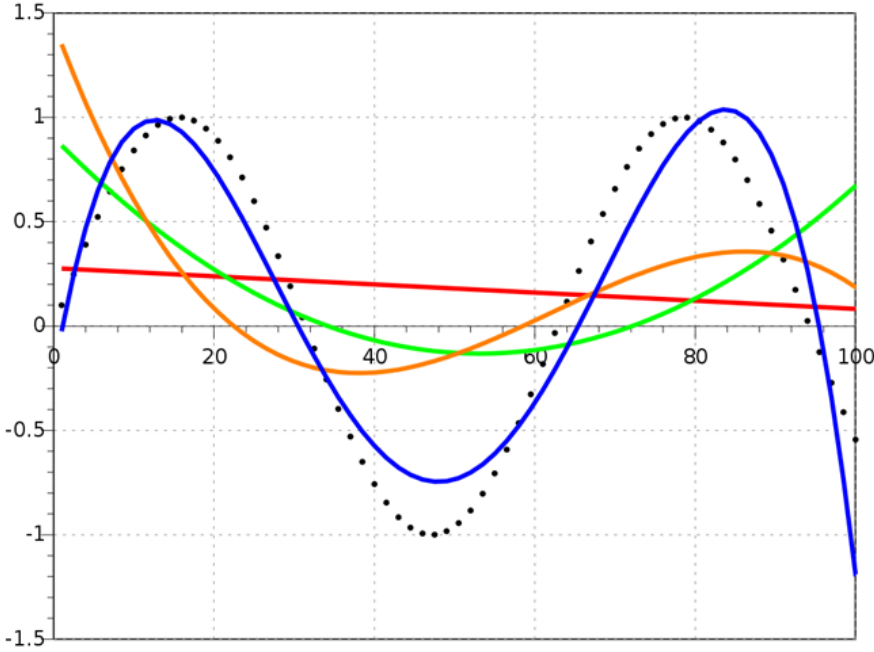
					ДП ІС-5119.1181-с.ССК						
					Схема структурна компонентів	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата	Комплекс задач аналізу ринку праці фахівців з інформаційних технологій	Аркуш 1		Аркушів 1			
Розробив		Пономаренко А.О.									
Перевірив		Звенігородський О.С.									
Т. кон.					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51						
Н. кон.		Москаленко Н.В.									
Затвердив		Звенігородський О.С.									



					ДП ІС-5119.1181-с.СБД			
					Схема бази даних	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Пономаренко А.О.						
Перевірів		Звенигородський О.С.						
Т. кон.						Аркуш 1		Аркушів 1
Н. кон.		Москаленко Н.В.			Комплекс задач аналізу ринку праці фахівців з інформаційних технологій	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51		
Затвердив		Звенигородський О.С.						

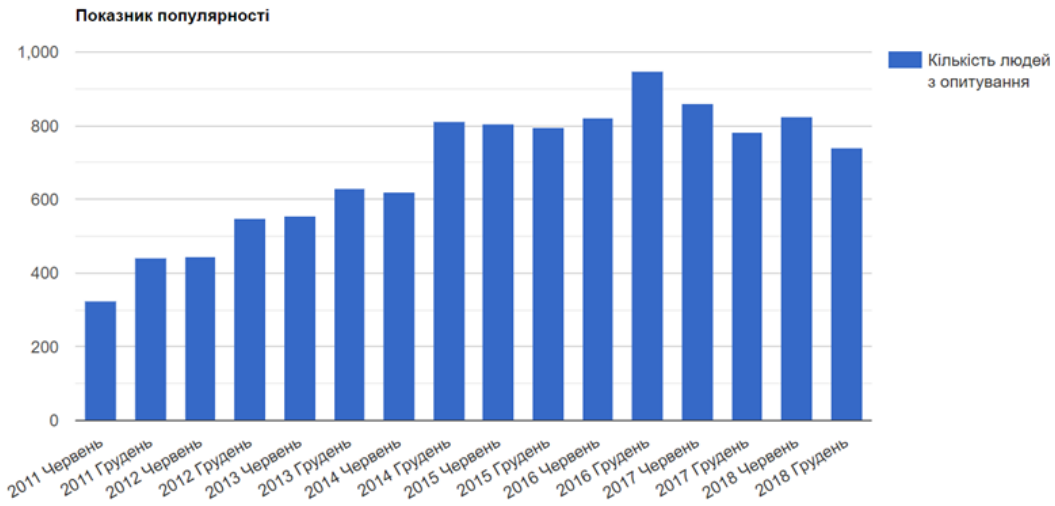


					ДП ІС-5119.1181-с.ССК					
					Схема структурна класів програмного забезпечення		Літера	Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата						
Розробив		Пономаренко А.О.								
Перевірів		Звенігородський О.С.								
Т. кон.						Аркуш 1		Аркушів 1		
					Комплекс задач аналізу ринку праці фахівців з інформаційних технологій		КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51			
Н. кон.		Москаленко Н.В.								
Затвердив		Звенігородський О.С.								

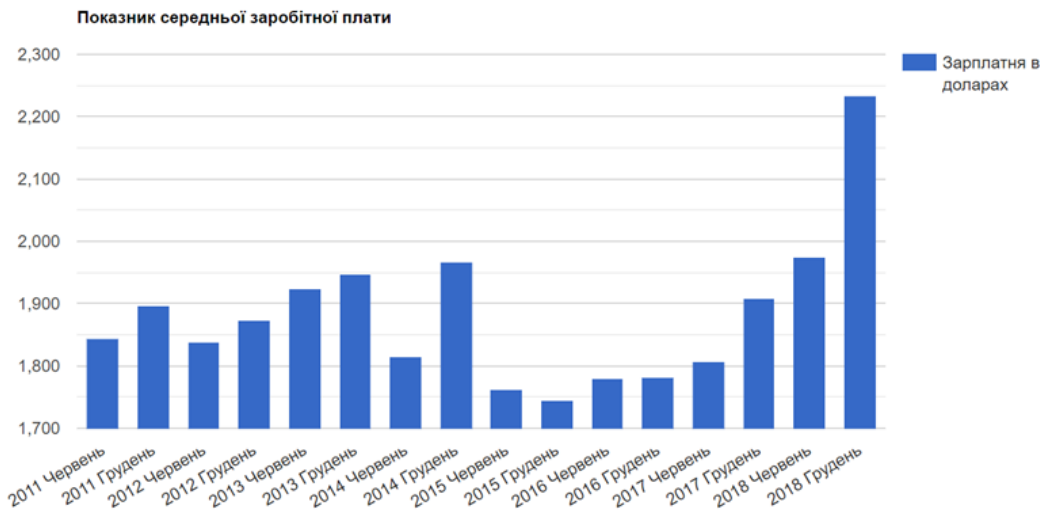


					ДП IC-5119.1181-с.РМЗ			
					Рішення з математичного забезпечення	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Пономаренко А.О.						
Перевірів		Звенигородський О.С.						
Т. кон.					Комплекс задач аналізу ринку праці фахівців з інформаційних технологій	Аркуш 1		Аркушів 1
Н. кон.		Москаленко Н.В.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. IC-51		
Затвердив		Звенигородський О.С.						

Місто	Посада	Мова програмування
<input type="text" value="вся Україна"/>	<input type="text" value="будь-яка"/>	<input type="text" value="будь-яка"/>
Рівень англійського		
<input type="text" value="будь-який"/>		
Рівень освіти	Досвід більший ніж:	Досвід менший ніж:
<input type="text" value="будь-який"/>	<input type="text" value="Менше року"/>	<input type="text" value="Більше десяти років"/>
<input type="button" value="Відправити запит"/>		



Місто	Посада	Мова програмування
<input type="text" value="всі Україна"/>	<input type="text" value="Software Engineer"/>	<input type="text" value="Java"/>
Рівень англійського		
<input type="text" value="будь-який"/>		
Рівень освіти	Досвід більший ніж:	Досвід менший ніж:
<input type="text" value="Высшее"/>	<input type="text" value="Менше року"/>	<input type="text" value="Більше десяти років"/>
<input type="button" value="Відправити запит"/>		



c = 1.0e+03 *

Columns 1 through 6

1.6713 1.8510 1.8270 1.7954 1.7884 1.8589

Columns 7 through 12

1.7789 1.7872 1.7087 1.6460 1.6617 1.6654

Columns 13 through 16

1.6773 1.7372 1.8400 1.9909

c2 = 2.0920e+03

					ДП ІС-5119.1181-с.КЕ		
					Креслення вигляду екранних форм		
Зм.	Арк.	№ документа	Підпис	Дата			
Розробив	Пономаренко А.О.						
Перевірів	Звенігородський О.С.						
Т. кон.							
					Літера	Маса	Масштаб
					Аркуш 1	Аркушів 1	
Н. кон.	Москаленко Н.В.				Комплекс задач аналізу ринку праці фахівців з інформаційних технологій	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-51	
Затвердив	Звенігородський О.С.						

Рішення з математичного забезпечення

Вхідні дані

Результати опитувань фахівців галузі ІТ за 2011 – 2018р в форматі csv: на сайті github.com [2]

Постановка задачі.

Дано. Часовий ряд рівня заробітної плати ІТ фахівців за півріччя (2011-2018р).

Зробити. Прогноз рівня заробітної плати на наступне півріччя за допомогою штучної нейронної мережі.

Підзадачі, які необхідно вирішити:

- вибрати ознаки вхідного вектора навчальної вибірки;
- вибрати тип і структуру штучної нейронної мережі;
- навчити нейронну мережу і емпірично підібрати параметри навчання.

Для нейронної мережі вибрані дві вхідні ознаки x_1 – рівень заробітної плати і x_2 – ознака зміни, якщо на попередньому кроці було зменшення, то $x_2=1$, якщо збільшення, $x_2=2$.

Таким чином загальна вибірка з даних сайту за 2011- 2018 роки виглядає наступним чином (для простоти моделювання півріччя позначені числами від 1 до 16:

- $x_1 = 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16$;
 - $x_2 = 2 \ 2 \ 1 \ 1 \ 1 \ 2 \ 1 \ 2 \ 1 \ 2 \ 1 \ 1 \ 2 \ 2 \ 2 \ 2$;
- $y = 1695 \ 1863 \ 1857 \ 1839 \ 1806 \ 1878 \ 1813 \ 1855 \ 1717 \ 1721 \ 1670 \ 1658 \ 1669 \ 1750 \ 1840 \ 2000$.

Результат прогнозування на наступне півріччя

$c = 1.0e+03 *$

Columns 1 through 6

1.6713 1.8510 1.8270 1.7954 1.7884 1.8589

Columns 7 through 12

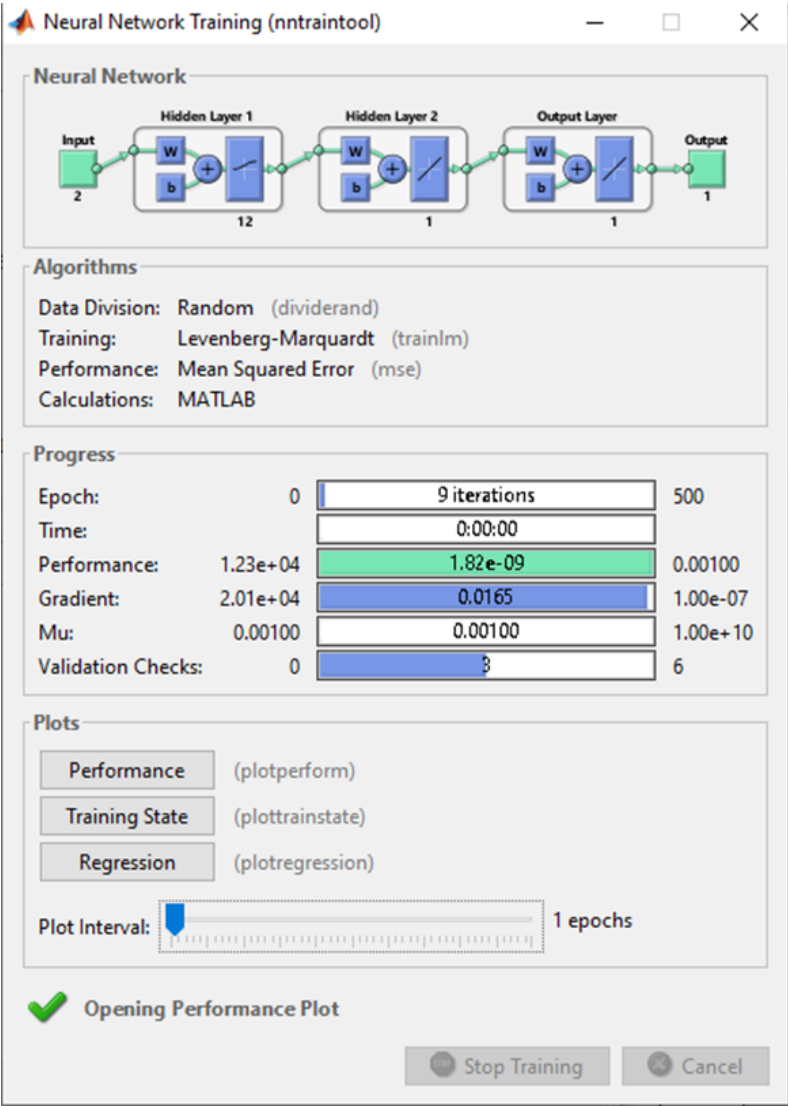
1.7789 1.7872 1.7087 1.6460 1.6617 1.6654

Columns 13 through 16

1.6773 1.7372 1.8400 1.9909

$c_2 = 2.0920e+03$

Мережа складається з трьох шарів. Перший шар містить дванадцять нейронів, другий та третій по одному. В першому шарі використовується сигмоїдна порогова функція, в інших – лінійна. Мережа з даною конфігурацією була навчена за 9 епох. Навчання було зупинене завчасно через досягнення необхідної точності результату.



Демонстраційний плакат до дипломного проекту

„Комплекс задач аналізу ринку праці фахівців з інформаційних технологій ”

Виконав студент гр. ІС-15

Пономаренко А.О.

Керівник ДП

Звенігородський О.С.